Better Modelling Practice

an ontological perspective on multidisciplinary,

model-based problem solving

Promotor:
Prof. ir. A.J.M. Beulens, hoogleraar Toegepaste Informatiekunde, Wageningen Universiteit

Co-promotor:
Prof. ir. M.S. Elzas, emeritus hoogleraar Kennissystemen, Wageningen Universiteit

Promotiecommissie:
Prof. dr. ir. J. Top, Vrije Universiteit, Amsterdam
Prof. dr. A.C. Smaal, Wageningen Universiteit, Wageningen
Dr. M. Makowski, International Institute of Applied Systems Analysis, Laxenburg, Austria
Prof. dr. ir. A.J. Udink ten Cate, Open Universiteit, Heerlen, C.H. Windesheim, Zwolle

Dit onderzoek is uitgevoerd binnen de onderzoekschool *Mansholt Graduate School of Social Sciences*

Better Modelling Practice

an ontological perspective on multidisciplinary,

model-based problem solving

Huub Scholten

Proefschrift
ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van Wageningen Universiteit
Prof. Dr. M. J. Kropff
in het openbaar te verdedigen
op vrijdag 19 december 2008
des voormiddags te 11.00 in de Aula.

Huub Scholten (2008). Better Modelling Practices: an ontological perspective on multidisciplinary, model-based problem solving. PhD thesis, Wageningen University – with references and with summaries in English and Dutch.

ISBN 978-90-8585-304-6

Keywords: knowledge base, ontology, mathematical model, modelling, problem and object system, quality assurance.

Table of Content

All chapters and appendices have more detailed Tables of Content.

1 Introduction

Quality management has much in common with sex:
everyone is for it, everyone feels they understand it,
everyone thinks execution is only a matter of following one's natural inclinations.
(Crosby, 1979)[1]

Content of Chapter 1

[1] Crosby P. (1979). *Quality is free*, McGraw-Hill Book Company, New York.

1.1 The credibility of model-based decision making

The world of science changed since the days of the '*uomo universalis'*, Leonardo da Vinci (1452-1519) and the Dutch brothers Christiaan and Constantijn Huygens (1629-1695, respectively 1628-1697), who bridged the canyon-sized gaps between art and science or replaced a philosophical foundation of science by a technological one. People like Da Vinci and the brothers Huygens have always been rare and life has become more complex. Solving problems relies to a large extent on science and scientific methodology and practices. Furthermore, problems are often too complex to be solved in the classical ways of Da Vinci or the brothers Huygens, by thinking or doing experiments. At present, problem solving is beyond the human dimension of renaissance scientists and requires tools that integrate and objectify scientific knowledge. These tools are called models and they try to represent scientific understanding of dominant processes in some system, in which the problem at hand is situated. Furthermore, models aim at supporting decision making in a qualitative or, preferably, quantitative way. The 'clients' of these scientifically based decisions can be other scientists, policy makers, business managers or interested members of the general public at large. For a large part of the public, models stand between science and some decision that affects them. Yet these groups are seldom familiar with models and typically do not understand models. If models subsequently support scientifically or socially 'wrong' decisions, one can imagine that science and its models are not commonly accepted instruments. In the eyes of many, modelling scientists belong to a group, in which Faust and Strangelove impersonate genial but evil descendants of Da Vinci and the brothers Huygens[2].

Decision practice in modern society requires scientifically sound and validated decision, design and planning processes for industry, policy making and management. At the same time many question marks are set at the models used to support these processes, at the quality of the decisions and at the decision processes, this situation being aggravated by a lack of transparency, not only for the public and the decision makers but also for other scientists.

The public confidence in models and modelling is decreasing and the public at large does not see models as reliable tools to support decision making. Due to a growing number of incidents related to decision making based on the outcomes of mathematical models, the credibility of modelling is declining. Modellers are aware of this situation that endangers modelling since almost 20 years. Many initiatives have been started to overcome this negative perception by decision and policy makers, stakeholders, journalists and concerned members of the public. Most of these initiatives choose for tackling the issue of uncertainty, which hinders straightforward use of model results. Policy and decision makers and their advisors do not like uncertainty. They have to find their way in the maze of opportunities between rational decision making and operational policy making.

Fifty years ago, when modelling was young and computers primitive, models were mainly used within the scientific community. Experts based decision support for policy making on qualitative reasoning. These experts pretended that they could cope with counteracting cause-effect relations and behavior of non-linear systems.

Thirty years later models were used as tools to provide quantitative advice. Often these models were mathematical models. In those days they were typically used without any estimation of uncertainty in the outcomes, simply because of the lack of methods to perform an uncertainty analysis and because decision makers did not like uncertain results. The decision makers had precisely hired the modelling whiz kids to get quantitative and thus exact advice.

In the last twenty years no panacea method to estimate model uncertainty has been developed. Due to the exponentially growing power of computers, rather primitive but robust methods like Monte Carlo analysis, are capable to solve at least parts of this problem. Decision makers do still not like uncertain model outcomes, but the growing number of modelling incidents prevents modellers and decision makers to ignore modelling problems. The Dutch 'modelgate' affair with most media exposure is called after its whistleblower De Kwaadsteniet. Triggered by a newspaper article[3], he started a technical discussion on statistical methods

---

[2] Haynes R (1994) *From Faust to Strangelove: Representations of the Scientist in Western Literature* Johns Hopkins University Press.

[3] This newspaper article has been published in the Dutch newspaper 'Trouw' on January 20, 1999.

and data for models used by the RIVM for its annually published report on environmental planning[4]. The discussion was focused on the absence of uncertainty ranges in the figures and tables in this report and has resulted in a significant increased awareness of uncertainty among scientists and policy makers.

Another affair in 'modelgate' is related to noise pollution caused by intensive air traffic affecting millions of people. Actual noise pollution should stay within limits set by the government. In many countries, including the Netherlands, actual air traffic noise pollution has to be calculated by a model instead of measured. In each country they use (forced by law) different models. Using a slightly different model can have significant impact on the model outcomes and associated economic (airport) and social (quality of life) impacts. Improper dealing with model uncertainties enables politicians to perceive the number of air traffic take-off's and landings as a political issue instead of a technical/scientific one[5].

Many other examples show how modern society is permeated with models and modelling. Another example of the ambiguous role that models can play in present day decision making can be found in the so called EVA-study, which aimed at unraveling the competition for bivalves like cockles, mussels and other shellfish and tidal zone benthic species in those parts of the Netherlands, where oystercatchers occur in high densities (being the Dutch part of the Waddensea and the Oosterschelde estuary). All (national) experts on the problem at hand have been consulted to some extent. The first part of the study, EVA I (1993-1997), was focused on the use of an oystercatcher model[6], which aimed at predicting the individual growth of oystercatchers, living in a population and showing feeding behavior in accordance with their social status in the population and given the measured but not modeled densities of their benthic feed (mainly cockles and to a lesser extent mussels and other benthic organisms)[7]. The model produced highly uncertain outcomes, resulting in an inconclusive policy advice. The variability in feed density was no model outcome so the effect of fishing cockles on the individual oystercatchers and the population was not adequately estimated in the modelling exercise. In EVA II[8] a new model has been developed for the Dutch part of the Waddensea and subsequently used for the Oosterschelde. Opposed to modelling in EVA I, where the model was thoroughly calibrated, preliminarily validated and the results presented with uncertainty estimates, the model in EVA II has not been calibrated and validated[9]. Besides the ostrich policy of leaving out those model analysis techniques that add nuances to the very precise model outcomes by providing estimates of the quality of the model results, the EVA II researchers made errors, which can be added to any list of modelling bloopers[10]. More recent studies estimated the total amount of harvestable cockle biomass in the Waddensea many times higher than the model predictions accounted for[11], leading to higher cockle fishery permits, despite the fact that the Dutch government has forbidden cockle fishery in Dutch coastal waters from 1 January 2005 on. The recent studies indicate that modelling of processes is difficult and risky, if the processes are not fully understood by scientists. In such cases – models based on insufficient and imperfect knowledge – modelling is used to provide a false scientific base to political decision-making.

The list of clashes between decision-making with and without models is endlessly long. At first sight, this looks like a paradox: models are usually the carrier of scientific research allowing the analysis of a problem, in which relevant knowledge on problem aspects are weighted quantitatively. Therefore studies based on

---

[4] 'Environmental Balance', published annually; in Dutch 'Milieubalans'.

[5] A.G.M. Dassen, J.H.J. Dolmans, J. Jabben, N.A.R. Hamminga, W.H. Hofmans, H.A. Nijland, 2000. *Geluid in de vijfde milieuverkenning- achtergronden*, RIVM rapport 408129 009, Bilthoven, Nederland (in Dutch).

[6] De Winter, W., and H. Scholten. 1997. Operationalisatie, calibratie, validatie en gebruik van het model EFFECT. Research Report of the EVA I project, Dept. Computer Science Wageningen University, Wageningen 150 pp.

[7] The (densities of) benthic species, on which oystercatchers feed, were not included in the model, because the inter-annual variability of these animals is not fully understood and therefore the densities are too difficult to model.

[8] Rappoldt, C., Ens, B., Berrevoets, C., Geurts van Kessel, A., Bult, T., Dijkman, E., 2003. Scholeksters en hun voedsel in de Oosterschelde; rapport voor deelproject D2 thema 1 van EVA II, de tweede fase van het evaluatieonderzoek naar de effecten van schelpdiervisserij op natuurwaarden in de Waddenzee en Oosterschelde 1999-2003. Alterra-Rapport 883. Alterra, Wageningen, 137 pp.

[9] Appendix C of Rappoldt *et al*., 2003, see the previous footnote.

[10] Many aspects of the model are not clearly described, e.g. food intake rates, especially those associated with other preys than the regular prey species (cockle, Baltic clam, mussel); more serious errors include (but are not restricted to): oystercatcher individual weights can decrease to values far beneath the weight an oystercatcher can survive.

[11] A short note in the Dutch newspaper 'De Volkskrant' of 29 July 2004, page 3, mentioned a governmental permit to catch 8 million kg of cockles in the autumn of 2004.

models should give better decision support. An emergent but preliminary conclusion is that modelling should prove its credibility.

Dynamic simulation models claim to mimic the behavior of some object system. The outcomes of these models are considered as the behavior of the modeled system in time. But is this true? A simple *yes* or *no* is often insufficient to answer this question, suggesting ambiguity in models and modelling. Some persons, considering themselves as modellers, often refer to modelling as the *art* of modelling, in which *art* refers to something unscientific. Why are scientists and decision makers still involved in modelling? The correctness of models and modelling should at least be questioned and investigated.

Can our society survive without models? In theory we will survive, but we will have to live with sub-optimal decisions or even wrong decisions, as many decisions are based on a modelling study. Therefore researching the correctness of models and their use is quite relevant. Since almost thirty years this type of research occurs, but its results are not very convincing. Pivotal issues are validation and quality, but both themes are fuzzy and vague by their permanent focus on model validity, a characteristic of models that can never be guaranteed. The negative view on modelling presented in this section is opposed by many positive developments in modelling. Therefore the focus should be: how to avoid modelling bloopers and improve modelling.

1.2 Problem solving, models, modelling and simulation

Before continuing with a discussion on the why and what of this book, some terms need to be explained here, including *model-based problem solving*, *models*, *modelling* and *simulation*.

If some problem owner has a problem on which sufficient knowledge is available, some modeller can develop or select a *conceptual model*, which contains this knowledge. Such conceptual model should contain relevant aspects of the problem and the contextual system in which the problem occurs, the so-called object system. These relevant aspects consist of dominant entities and processes. Linking the conceptual model with data observed in reality transfers the conceptual model into a *model instance* that can help to solve the actual problem.

'Modelling' refers to developing and/or using models in order to solve some problem, in other words *model-based problem solving*. The models discussed here can be categorized as mathematical models, as they share the use of mathematics to represent relevant parts of an object system and to solve the problem at hand. The models may range from discrete or continuous simulation models to operations research models. Simulation models aim at mimicking object system's behavior, often by minimizing some predefined measure for differences between model outcomes and observations in the object system (OS). Many other mathematical models try to reach their goal by similar optimization techniques. Mathematical modelling is a major building block of this book and will be introduced in Chapter2.

Model-based problem solving or modelling can be seen as a process consisting of many things to do, e.g. *problem formulation*, *data handling*, *model implementation*, *sensitivity analysis*, *model calibration*, *uncertainty analysis*. Defining this modelling process in detail can provide a shared concept on how to model, shared by larger parts of the modelling community. If a modelling team runs a model study according to the modelling definition and keeps records of what team members actually do, this will make the modelling process more transparent. The modelling team records can serve as an audit trail and include all assumptions, activities, results and choices/decisions made by the team. In addition to a modelling definition and records of a modelling study, explicit descriptions of *models* and of the combination *problem and its objective system* are needed. Explicit descriptions of model, problem / object system and modelling should represent the shared vision of a team. This book assumes that these explicit descriptions together with the modelling team records can act as cornerstones of a framework to improve the use of models in model-based problem solving.

At present, many use the compound term 'modelling and simulation (M&S). In the latter, 'modelling' refers to building or adapting a model for a certain object system and problem situation and 'simulation' to mimicking the behavior of this object system by running the model. In the context of the cases used in this book ‘simulation’ is not frequently used, opposite to the world of (professional) system scientists, where it is very popular. Chapter 2 will provide more technical details on mathematical model and modelling.

1.3 My personal involvement

When I was an undergraduate student in the 70’s of last century, I found my first essential errors in mathematical models published in refereed journals in the field of freshwater ecology. I was astonished. Several years later, I met a manager of a well-known modelling group in a train in southern Great Britain both traveling to a workshop where we were expected to give presentations on our respective ecosystem models. He claimed that his (large and complex) ecosystem model was without programming errors. I wondered that even a well-known scientist and modeller as he was, could be so sure of something so vulnerable for smaller and bigger mistakes. But he was rather convinced of his statement, which was rephrased and published as follows: “*The code has been verified to ensure that all programming errors have been removed*.” (Gordon *et al*., 1986).

Between 1985 and 1995 I spent my time developing models and modelling software. A major issue at that time in ecosystem modelling was how to deal with the gigantic uncertainties in model outcomes. One of the causes is data uncertainty associated with the state of the chemical analysis technology at that time, conceptual problems and the very hard conditions under which the data have to be collected. Gaps in the body of knowledge of the dominant processes involved in ecosystem modelling are perhaps a more important cause of model outcome uncertainty. Because of this my personal interest shifted in these years from model development (Klepper *et al.*, 1994), to model calibration, uncertainty analysis and model validation issues (Klepper *et al.*, 1991, Scholten and Van der Tol, 1994, 1998). My interest in model uncertainty and how to deal with that in calibration and uncertainty analysis has been inspired by the work of the group centered around Gardner and O'Neill (O'Neill and Gardner, 1979, O'Neill and Rust, 1979, O'Neill *et al*., 1980, Gardner *et al.*, 1981, 1982, O'Neill *et al.*, 1982a, 1982,b, Gardner and O'Neill, 1983, Gardner *et al.*, 1989). My interest in model validation was partly triggered by the same group (Mankin *et al.*, 1975, Caswell, 1976), but the validation issue has also been discussed by others (Hermann, 1967, Wigan, 1972, Lewandowski, 1982). Since twenty years most publications on validation issues are authored by Sargent (1982, 1984a, 1984b, 1986, 1989a, 1989b, 1999, 2003), but many other authors have contributed to this topic too (e.g. Young, 1983). See for some reviews Knepell and Arangno (1993), Walker *et al*., 2003 and Refsgaard and Henriksen, 2004.

In 1994 I perceived a need for improving modelling beyond a validation perspective. My interest was fuelled by several publications in national newspapers and (grey) reports on guidelines for air pollution modelling from a modeller’s viewpoint[12] or from a software engineer’s perspective[13]. Others exposed the need for model-based water management guidelines[14]. Sargent’s view on validation is broader than of most authors on validation, including conceptual validation, data validation, software verification, historical data reproduction and operational validation. But even model studies passing all these validation tests, appear often error-prone and result in incorrect or inadequate results. This perception motivated me to look at the research topic of quality assurance in (dynamic) modelling and simulation (Scholten, 1994, Scholten and Udink ten Cate, 1995, 1996, 1999). Due to this interest I was asked to join the team involved in the realization of a Good Modelling Practice Handbook for water management (Scholten, 1999, Van Waveren *et al.*, 1999, Scholten, *et al.*, 2000, Scholten, 2001, Scholten and Osinga, 2001, Scholten and Groot, 2002). In a formalization step this GMP Handbook has been upgraded into a set of official Dutch norms to be used for model-based water management[15].

[12] Dekker, C.M., Groenendijk, A., Sliggers, C.J., 1990. Kwaliteitscriteria voor modellen om luchtverontreiniging te meten. Ministerie van Volkshuisvesting en Ruimtelijke Ordening, Den Haag, 52 pp.
[13] Dekker, C.M., 1991. Inzet rekenmodellen vergt criteria voor kwaliteit. Computable, 39-41.
[14] Van der Giessen, A., De Haan, B.J., Steinberger, P.E., 1992. Kwaliteitsborging van wiskundige modellen. RIVM, Bilthoven, 29 pp.
[15] At present the set of norms consists of Design Norm NEN 6260-1: *Termen en definities van modelleren* (in English: Terms and definitions of modelling), Design Norm NEN 6260-2: *Basisprincipes van modelleren* (in English: Basic principles of modelling) and Design Practice Guideline NEN 6260-3: *Valkuilen gevoeligheden van modelleren* (in English: Pitfalls and sensitivities of modelling). A Practice Guideline has a lower level of maturity and is less formal than a Norm.

The wide interest in the GMP Handbook and the set of norms motivated me to initiate a research and development project, HarmoniQuA. The project was partly funded by the European Commission and aims at assuring the quality of model-based water management[16]. A large part of this book is a reflection of lessons learned in HarmoniQuA and many of the concepts and ideas are results of the work and discussions in this project.

My personal interest fits in the research focus of the group, in which I work. The group has more than 25 years of experience in developing tools, methods and concepts for modelling, and has participated in defining the theoretical concepts of this paradigm. It has published many books and a very large number of research papers and reports, organized workshops and conferences, and developed several (software) platforms for modelling. It co-operates with many research groups, both within the applied field of building models and in more methodological domains, within the university, nationally and internationally. The models cover a broad range of application domains, including model based environmental decision support systems for aquatic ecosystems and water management problems in general. Besides models, the group has a wide expertise in developing solvers for continuous, discrete and hybrid simulation models, as well as in model analysis tools for calibration and of models with stochastic parameters. The group has a leading role in defining and arranging knowledge on water management in formal knowledge bases using state-of-the-art, ontology based technology and in building tools to interact with this knowledge.

1.4 What kind of problems will be addressed

Most initiatives to overcome problems related with modelling incidents in the Netherlands and in several other countries, lead to an increased interest in model related and other uncertainty, but there are many other lessons learned resulting in a variety of other approaches. Refsgaard (2002) summarizes modelling problems as follows:

- Ambiguous terminology and a lack of mutual understanding between key-players;
- Bad practice (careless handling of input data, inadequate model set-up, insufficient calibration/validation and model use outside of its scope);
- Lack of data or poor quality of available data.
- Insufficient knowledge on the processes hindering ecological (biota) modelling.
- Miscommunication of the modeller to the end-user on the possibilities and limitations of the modelling project and overselling of model capabilities;
- Confusion on how to use model results in decision making;
- Lack of documentation and transparency of the modelling process, leading to projects, which hardly can be audited or reconstructed.
- Insufficient consideration of socio-economic, institutional and political issues and a lack of integrated modelling.

Beulens and Scholten (2004) add two other problems to this list:

- In the acquisition phase the problem owner has to select a consultant or other organization that has the right expertise for the problem to be solved. In this tendering procedure the problem owner can make a complete wrong choice for a problem solver. This can cause various problems, including inadequate expertise to solve the problem.
- In the project start-up major selections (which problem related process to include and which type of model to use) can put the problem solving process on a complete wrong track.

Only a few of these problems are associated with uncertainty and validation issues, which is the topic of a large part of the scientific modelling community in their efforts to improve model credibility. The rest is associated with other aspects of the process of modelling.

What can go wrong in the model-based problem solving for water management can perfectly be illustrated with anecdotal, modelling 'bloopers' that actually occurred, which are obtained from the Good Modelling Practice Handbook (Van Waveren *et al.*, 1999). These anecdotal modelling errors will be summarized here.

- *Leaving out essential processes* and compensating for this by introducing a compensating factor in model calibration to get comprehensible results.

[16] HarmoniQuA contract EVK1-CT2001-00097, see www.HarmoniQuA.org/.

- *Erroneous assumptions* underlying the model, resulting in a 100% oversized retention basin.
- *Uncertainty* in model outcomes and measuring errors too large prohibiting the use of the model as a management instrument for compensating extra salt discharges.
- *Misinterpretation* of model calibration outcomes, resulting in a model describing system behavior, in which water was extracted from the sea, flowed up the mountain and disappeared on its top.
- *Using the wrong type of model*: a 1D hydrodynamics model instead of a 2D model, resulting in ignoring fivefold flow rate differences in the lateral dimension of a channel, which resulted in too high flow rates for ship passage under a new bridge.
- *Miscommunication* of model capabilities after improving a hydrodynamic model used for predicting water levels in a residential area close to a river. The old model version always overestimated winter water levels by 50 cm and the new model version was accurate. The outcomes of the new model version were interpreted as the results of the old version. The residential area was flooded at the moment the new model predicted this, leading to substantial damage.
- Due to *using an incorrect model solver* a water quality model produced negative nitrate and ammonium concentrations.
- An ecosystem model, unable to deal properly with *discontinuities* could not handle an adsorption mechanism with a threshold for ammonium by algae, leading to a persisting nitrogen balance.
- *Inadequate spatial modelling* (schematization) may trigger a model to underestimate a process of two alternatives, leading to an underestimated effect of the underestimated process, which only could be compensated during calibration by over estimating parameters associated with the underestimated process.
- *Illogical use of available knowledge*; incompatible time constants in water transport and growth of algae, hindering the latter process.
- *Operational errors* in model handling; some, obviously inadequate, solvers of differential equations cannot handle initial values being zero.
- Not being aware of the consequences of *leaving out some of the feedback mechanisms*; using forcing functions for system entities with impact on other system entities can lead to miscalculation of entities which are included in the model as state variable.
- *Incorrect or inconsistent use of units* in models/equations.

Although these examples originate from water management, it's easy to find similar erroneous model applications in other fields. Scholten *et al.* (2001) summarizes water domain independent sensitivities and pitfalls to avoid, which have been presented in Van Waveren *et al.* (1999). Many of these general pitfalls and sensitivities in modelling within the continuous simulation paradigm can be avoided when using these *lessons learned* (Scholten *et al.*, 2001):

- Monitor the modelling process: this makes model based studies easier to reproduce;
- Define and specify in an early stage:
  - what type of model outcomes are needed, especially in the case of chains of models;
  - the size of the area in view of the boundaries;
  - spatial and temporal resolutions (scales) in view of the problem at hand and for the choice of the model in view of the required functionality;
  - criteria for the model accuracy;
- Harmonize the manager's model objectives with its technical translation;
- Define a sound conceptual model with a proper choice of processes: errors with this will easily be masked in a calibration;
- Detail a model at a level which harmonizes with the available data;
- Choose the appropriate software for the problem at hand;
- Allocate sufficient resources (time) for model analysis, i.e. sensitivity analysis, calibration, uncertainty analysis;
- Harmonize the number and choice of model parameters to calibrate a model with the available data and its measuring frequency, using the theoretical concepts of observability;
- Make a distinction between insensitive and sensitive parameters in calibration and calibrate the model only by using the latter;
- Be cautious in comparing point observations with model outcomes representing an area or a volume in any model evaluation;

- Be cautious in comparing observations averaged over time with model outcomes representing points in time in any model evaluation;
- Do not believe the model is good since the calibration is perfect;
- Use a model within its scope;
- Use sufficient run-in time before the actual simulation period;
- Account for output uncertainty (ranges, distributions) when comparing different scenarios;
- Use understandable terminology in communicating the results to clients (decision makers);
- Explain the results verbally in addition to written reports.

This collection of *lessons learned* deal with various problems. Some have to do with proper following the guidance provided by a shared definition of the modelling process, some with communication between members of the modelling team, some with availability and quality of data, which allow to instantiate and use a specific model type for the case, some with model fit, some with the predictive power of the model and some with translation of modelling slang to terminology for decision makers. In brief, the lessons learned refer to modelling as process, to model type and content and to adequately structuring relevant knowledge on problem and object system.

Some general remarks have to be made on model-based problem solving. If a *problem owner* has a problem to be solved, this problem is always a specific problem, i.e. a problem occurring at a specific place and time and within a specific and realistic context. Such specific problem is not well defined yet and usually it belongs to some known or unknown class of more generic problems. Problems[17] are solved using *problem solving methods*, which belong to a specific *problem solving paradigm*. Many problem solving paradigms use mathematical models in the problem solving process. In order to determine how well a *model-based problem solving paradigm* can solve a specific problem two new concepts have to be introduced, related to the ability of mathematical models: *representation demand* and *representation power* (Beulens and Scholten, 2001). A specific problem has representation demands and a model-based problem solving paradigm has representation power[18]. Differences between representation demand and representation power can provide information on how to continue in the process of problem solving (see Figure 1-1). If the representation power of a model-based problem solving paradigm is insufficient to represent a specific problem, the model-based problem solving paradigm cannot fully solve this problem. The problem owner should decide whether to accept this or to choose another or complementary approach by switching to one or more other model-based problem solving paradigms or to go for a multidisciplinary problem solving process.

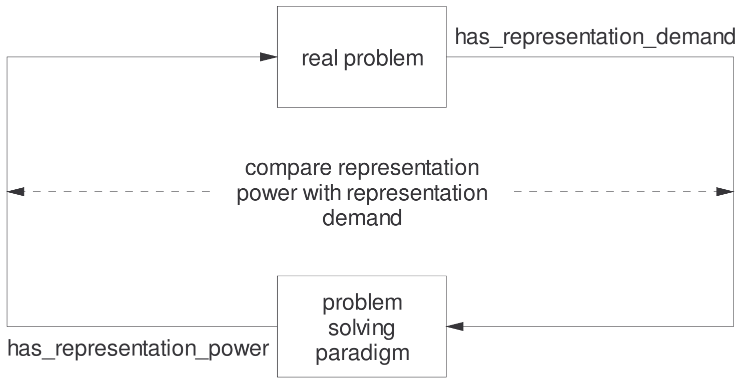


Figure 1-1. How to solve a problem.

At present many real problems, suited for a model-based problem solving approach, are too complex for a monodisciplinary approach. Often one tries to solve this type of problems by choosing a monodisciplinary approach. Multidisciplinary models have extra difficulties to overcome, often related to communication clashes in the modelling team, conceptual incompatibilities and technical hurdles at model solver level or at software level.

---

[17] A *problem* is a situation asking for understanding or for a way (i.e. solution) to change an unwanted (present) situation in a wanted (future) situation. The unwanted situation can be characterized by an obstacle that has to be removed or by being far from a desired, ideal state of affairs.
[18] The *representation power* also depends directly on the expertise of the modeller.

1.5 Research questions

The basic question, fuelling all research effort in this book, can be summarized as follows: *How to improve the quality of modelling in order to increase its credibility.* This statement is obviously too vague to be used as a research question. Therefore the problems outlined in section 1.4 need to be translated into tangible research objectives and a methodological pathway to this these goals.

A first step to make the basic question more explicit is to split it into three objectives:

1. The *modelling* process[19] should be executed according to 'good modelling practices';
2. The *model* should be scientifically 'good'[20] and 'useful'[21]*;*
3. The model can only be 'good' and 'useful', if it is based on adequate explicit descriptions of the *problem* at hand and its associated *object system*.

The first objective – good modelling practices – comprises the second, which is related to the quality of the model, but the first consists furthermore of all pieces of the complex process of modelling (model development, analysis and application). Good practices are based on consensus and a shared view of the professional community on how to apply good practices. The good practices can be described by defining modelling in terms of what has to be done. These good modelling practices should provide guidance to modelling teams. In addition, this guidance on what to do should be accompanied by records of what actually has been done. These records can be used to manage a modelling project and act as audit trail. The audit trail allows reconstructions of modelling projects and enhances their transparency.

The second objective – a good and useful model – is strongly interdependent with the other two objectives. The linkage between the second and the first objective is obvious, as making a good and useful model is part of the modelling process. The linkage between the second and the third objective can best be explained as follows. Good and useful models should represent those structural elements of reality that are relevant for the problem at hand. Furthermore – in case of a dynamic simulation model – a model should be able to produce output that shows a similar behavior in time as the relevant entities of reality. Therefore an object system should be defined, which contains and describes – in a structural way – those entities and processes that govern aspects of reality, relevant for the problem at hand. Finally, a good model requires a sound scientific base, i.e. the model assumptions, structure and process included should be in accordance with state-of-the-art scientific achievements.

The third objective – a well and explicitly described problem and object system – is required for two reasons. Firstly, an appropriate description of the problem at hand helps solving it. Secondly, an explicitly and aptly defined object system is a prerequisite to choose or develop a good and useful model.

Now, the basic question '*how to improve the quality of modelling in order to increase its credibility*' and its translation into objectives (good modelling practice, good and useful models, and an appropriate description of problem and object system) can be translated into research questions that will help to achieve these objectives:

1. How can a *problem* and its associated *object system* be described in a way that enables the selection, development or instantiation of an appropriate model to solve the problem at hand?
2. How should a *model* be described in a way that makes it scientifically sound and instrumental to solve the problem at hand?
3. How should *modelling* be done according to good modelling practices and in a transparent way, allowing reconstruction of model-based problem solving projects?

---

[19] *Modelling* is here used for the process of choosing an appropriate type of conceptual model, making a model instance of it adding appropriate data about the problem and object system, analyzing it and applying it to the problem at hand, all according to a shared modelling definition accepted by a substantial part of the modelling community in that domain. Furthermore, communication within the modelling team, including the problem owner, stakeholders and auditors/reviewers, on how good the model is and how well the model is used, is an essential prerequisite.
[20] The word *good* is used as container term summarizing many positive qualifications, including but not limited to correctly representing the problem and object system in terms of structure and behavior.
[21] The term *useful* indicates that the purpose of a model here is solving the problem.

This book aims at improving the credibility of models and modelling along these lines. Models and modelling will never be perfect, but if their quality can be evaluated and their transparency established, this will improve the credibility. Instead of a classic approach in modelling (see Figure 1-2) a new approach will be proposed in this book (see Figure 1-3), which includes a framework with knowledge on modelling, models, problem and object system.

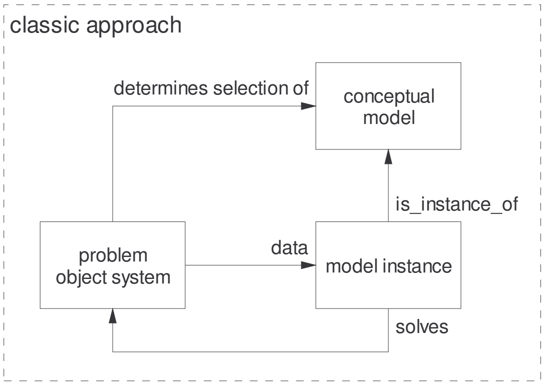


Figure 1-2. Classic approach to modelling.

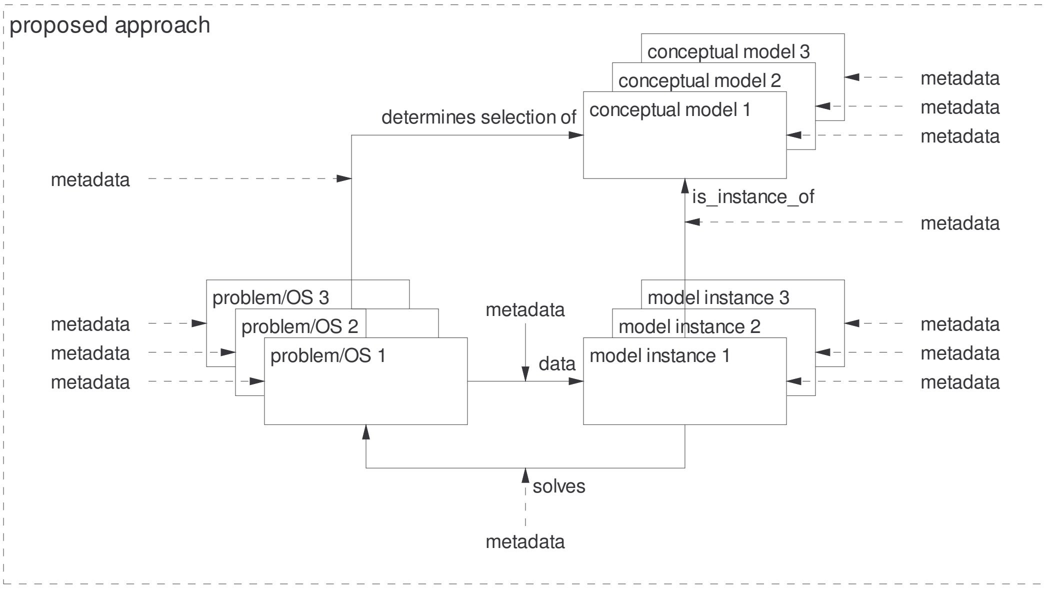


Figure 1-3. Proposed modelling approach. The different versions of conceptual models and model instances can be alternative formulations within a modelling paradigm or completely different types of models. OS means Object System. The arrows represent parts of the modelling process. Metadata refers to knowledge on *modelling*, *model*, *problem* and *object system*.

1.6 Outline of the research

The first ideas to do the research described in this book go back to 1994. The need for quality assurance has been clearly expressed in Scholten (1994). The first chance to achieve quality assurance requirements for model-based problem solving was with the development of a Good Modelling Practice Handbook between 1998 and 1999 (Van Waveren *et al*., 1999, Scholten, 1999, Scholten *et al*. 2000, 2001a, b). The GMP Handbook covered only partly what was outlined in Scholten (1994) and Scholten and Udink ten Cate (1999), as it was merely a guideline for modelling in water management and did not support modelling in a practical sense. It helped modellers in documenting what they did, but its structure (linear) and format (a book on paper) did not invite modellers to use it. Although its use has sometimes been enforced by clients of a model study (problem owners), the intrinsic qualities of the GMP Handbook itself hardly convinced anyone to use it. In a follow-up project the guidance provided by the book has been upgraded to a set of Dutch norms, which are more precise and consistent. But even perfect norms do not solve the real problem. Models and modelling will still have a bad reputation and a low credibility in certain domains. Other requirements include a theoretical framework and an operational framework for multidisciplinary model-based problem solving, together with a wide acceptance of the proposed methodology.

Between 1994 and the completion of the GMP Handbook, I was involved in modelling itself, mainly models to catch up with the eco-physiological response of bivalve filter feeders on changing feeding conditions, a detail of some ecosystem models. Furthermore I worked on establishing modelling software, which reflects the findings of theoretical modelling and simulation research. In the mid-nineties I discovered the power of ontologies to structure knowledge outside the reach of previous artificial intelligence knowledge structuring methods. This mixture of needs for improving modelling, personal modelling experience and being familiar with theoretical aspects of modelling and knowledge engineering technology allowed me to aim for an innovative approach, which effectively and efficiently helps modellers in their professional tasks. No straitjacket but a tailored tool facilitating better modelling.

In this book I will follow a trail from contributing disciplines, via philosophy of science to the theoretical and operational framework that aims at a full support of multidisciplinary model-based problem solving. This framework consists of four coherent and ontological knowledge bases containing basic terminology (meta-ontology), concepts for modelling (modelling knowledge base), concepts for problems and associated object systems (problem knowledge base) and concepts for models (model knowledge base). Furthermore, some tools aim to support developing the ontological knowledge bases and supporting multidisciplinary teams in collaborative model-based problem solving. The proposed framework is outlined in Figure 1-4.

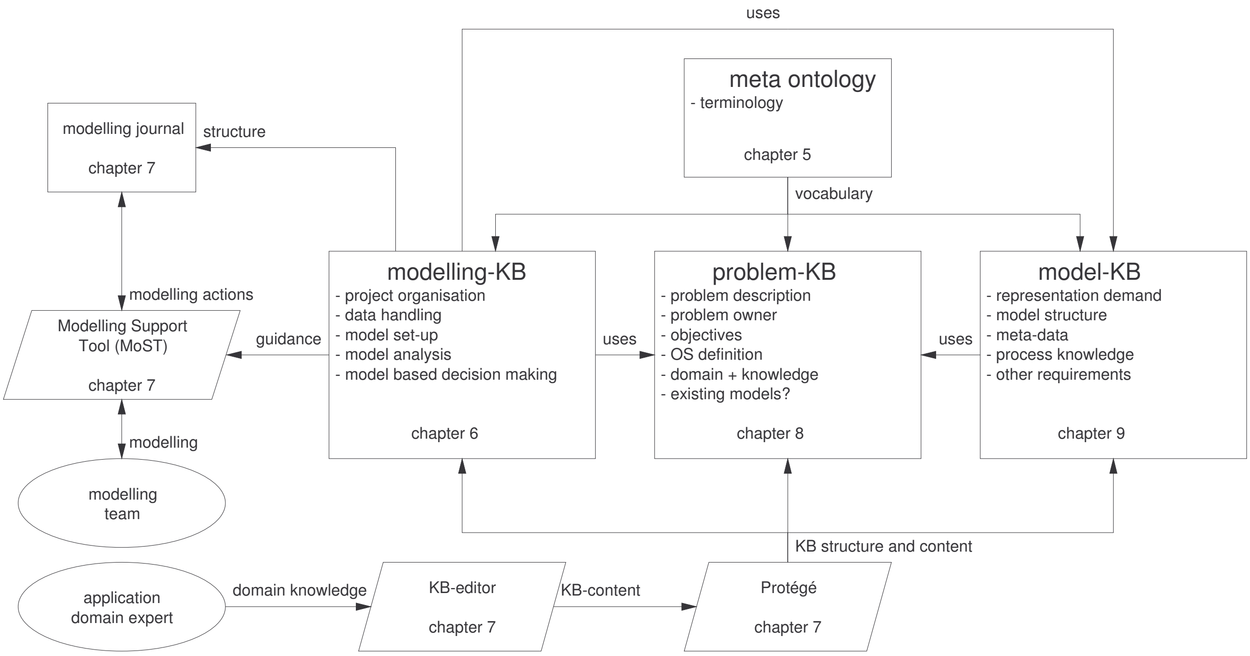


Figure 1-4. Outline of the proposed framework to improve modelling. Rectangles represent ontological knowledge bases, parallelograms represent (software) tools and ovals represent (groups of) persons.

1.7 Structure of the book

The approach presented in this book is based on many scientific achievements in the past, belonging to many paradigms. These methodological building blocks will be discussed briefly in Chapter 2. Chapter 3 introduces what ontologies are, why ontologies can help structuring knowledge, which is difficult to organize, and finally how to use ontologies for knowledge management. Chapter 4 aims at setting up a theoretical framework for multidisciplinary model-based problem solving. Chapter 5 fills a terminological gap, as it aims to bootstrap the definition of subsequent ontologies. In Chapter 6 a modelling ontology is proposed that can be extended for other processes too. Chapter 7 describes a software tool to support the modelling process. The next two chapters propose ontologies for problem and object system related knowledge (Chapter 8) and model representation (Chapter 9). In Chapter 10 the framework for multidisciplinary model-based problem solving will be evaluated. Chapter 11, the last chapter, discusses the methodological framework and associated tool.

1.8 References

Beulens, A.J.M. and H. Scholten, 2001, Challenges to model solving software in the Internet era. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering., Delft, The Netherlands, June 26-29, 2001, TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0.

Beulens, A.J.M., Scholten, H., 2002. Ontologies to structure models and modeling tasks. In: Makowski, M. (Ed.), Extended abstract of 16th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, 15-17 July, 2002. IIASA, Laxenburg, Austria, pp. 3-5.

Beulens, A.J.M., Scholten, H., 2004. An ontological framework for structuring process knowledge specified for the process of model-based problem solving, 18th Workshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment, 6 – 8 September, 2004. IIASA, Laxenburg, Austria, pp. 4-8.

Caswell, H., 1976. The validation problem. In: Patten, B.C. (Ed.), System analysis and simulation in ecology. Academic Press, New York, pp. 313-325.

Gardner, R.H., Cale, W.G., O'Neill, R.V., 1982. Robust analysis of aggregation error. Ecology 63, 1771-1779.

Gardner, R.H., Dale, V.H., O'Neill, R.V., 1989. Error propagation and uncertainty in process modelling, Proceedings of Conference on Forest Growth. Timber Press, Auburn University.

Gardner, R.H., O'Neill, R.V., 1983. Parameter uncertainty and model predictions: a review of Monte Carlo results. In: Beck, M.B., Van Straten, G. (Eds.), Uncertainty and forecasting of water quality. Springer Verlag, New York, pp. 245-257.

Gardner, R.H., O'Neill, R.V., Mankin, J.B., Carney, J.H., 1981. A comparison of sensitivity analysis and error analysis based on a stream ecosystem model. Ecol. Modelling 12, 173-190.

Gordon, D.C., Keizer, P.D., Daborn, G.R., Schwinghammer, P., Silvert, W.L., 1986. Adventures in holistic ecosystem modelling: the Cumberland Basin ecosystem model. Neth. J. Sea Res. 20, 325-335.

Hermann, C.F., 1967. Validation problems in games and simulation with special reference to models of international politics. Behavior Science 12, 216-231.

Klepper, O., H. Scholten, and J. P. G. Van de Kamer. 1991. Prediction uncertainty in an ecological model of the Oosterschelde estuary, S.W. Netherlands. Journal of Forecasting 10:191-209.

Klepper, O., M.W.M. Van der Tol, H. Scholten and P.M.J. Herman, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.nepell, P.L., Arangno, D.C., 1993. Simulation validation: a confidence assesment methodology. IEEE Computer Society Press, Los Alamitos, CA, 150 pp.

Lewandowski, A., 1982. Issues in model validation. Angewandte Systemanalyse 3, 2-11.

Mankin, J.B., O'Neill, R.V., Shugart, H.H., Rust, B.W., 1975. The importance of validation in ecosystem analysis. In: Innis, G.S. (Ed.), New directions in the analysis of ecological systems, Part 1. Simulation Councils Inc., Lajolle, California, USA, pp. 63-71.

O'Neill, R.V., Gardner, R.H., 1979. Sources of uncertainty in ecological models. In: Zeigler, B.P., Elzas, M.S., Klir, G.J., Oren, T.I. (Eds.), Methodology in systems modelling and simulation. North-Holland Publ. Co., Amsterdam, pp. 447-463.

O'Neill, R.V., Gardner, R.H., Barnthouse, L.W., Suter, G.W., Hildebrand, S.G., Gehrs, C.W., 1982b. Ecosystem risk analysis: a new methodology. Environmental Toxicology and Chemistry 1, 167-177.

O'Neill, R.V., Gardner, R.H., Carney, J.H., 1982a. Parameter constraints in a stream ecosystem model: incorporation of a priori information in Monte Carlo error analysis. Ecol. Modelling 16, 51-65.

O'Neill, R.V., Gardner, R.H., Mankin, J.B., 1980. Analysis of parameter error in a nonlinear model. Ecol. Modelling 8, 297-311.

O'Neill, R.V., Rust, B., 1979. Aggregation error in ecological models. Ecol. Modelling 7, 91-105.

Refsgaard, J.C. (ed.) (2002). State-of-the-art report on quality assurance in modelling related to river basin management. HarmoniQuA Report, D-WP1-1, Copenhagen, Denmark, www.HarmoniQuA.org/deliverables.htm.

Refsgaard, J.C., Henriksen, H.J., 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.

Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.

Sargent, R.G., 1982. Verification and validation of simulation models. In: Cellier, F.E. (Ed.), Progress in modelling and simulation. Academic Press, London, etc, pp. 159-169.

Sargent, R.G., 1984a. A tutorial on verification and validation of simulation models. In: Sheppard, S., Pooch, U., Pegden, D. (Eds.), Proceedings of the 1984 winter simulation conference, pp. 115-121.

Sargent, R.G., 1984b. Simulation model validation. In: Oren, T.I., Zeigler, B.P., Elzas, M.S. (Eds.), Simulation and model-based methodologies: an integrative view. Springer Verlag, Berlin, etc., pp. 537-555.

Sargent, R.G., 1986. An exploration of possibilities for expert aids in model validation. In: Elzas, M.S., Ören, T.I., Zeigler, B.P. (Eds.), Modelling and simulation methodology in the artificial intelligence era. Elsevier, Amsterdam, pp. 19-40.

Sargent, R.G., 1989a. Validation of simulation models: general approach. In: Singh, M.G. (Ed.), Systems and control encyclopedia: theory, technology, applications. Pergamon Press, Oxford, etc., pp. 5008-5011.

Sargent, R.G., 1989b. Validation of simulation models: statistical approach. In: Singh, M.G. (Ed.), Systems and control encyclopedia: theory, technology, applications. Pergamon Press, Oxford, etc., pp. 5015-5019.

Sargent, R.G., 1999. Validation and verification of simulation models. In: Farrington, P., Nembhard, H., Sturrock, D., Evans, E. (Eds.), Winter Simulation Conference, pp. 39-48.

Sargent, R.G., 2003. Verification and validation of simulation models. In: S. Chick, P.J. Sanches, D. Ferrin, Morice, D.J. (Eds.), Proceedings of the 2003 Winter Simulation Conference, pp. 37-48.

Scholten, H., 1994. Quality assurance of the simulation modeling process. In: Kaylan, A.R., Lehmann, A., Ören, T.I. (Eds.), Proc. of the European Simulation Symposium, Istanbul, Octobre 9-12, 1994. The Society for Computer Simulation International, Istanbul, pp. 85-89.

Scholten, H., 1999. Good Modelling Practice. In: Makowski, M. (Ed.), Abstracts of the 13th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment. IIASA, Laxenburg, pp. 57-59.

Scholten, H., 2001. An ontological approach in quality assessment of modelling and simulation. In: Heemink, A.W., Dekker, L., Arons, H.d.S., Smit, I., Stijn, T.L.v. (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering. TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0, Delft, The Netherlands.

Scholten, H., Groot, S., 2002. Dutch guidelines. In: Refsgaard, J. (Ed.), State-of-the-art report on Quality Assurance in modelling related to river basin management. GEUS, Copenhagen, pp. ch.12, 11-10.

Scholten, H., Osinga, S.A., 2001. How to support GMP in model-based DSS. In: Makowski, M. (Ed.), Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, August 27-29, 2001. IIASA, Laxenburg, Austria, pp. 45-47.

Scholten, H., R. H. Van Waveren, S. Groot, F. Van Geer, J. H. M. Wösten, R. D. Koeze, and J. J. Noort (2001). Improving the quality of model based decision support: Good Modelling Practice in water management. Pages 223-230 in: A. Schumann, J. Xia, M. Marino, and D. Rosbjerg (Eds). Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the IAHS, Maastricht, The Netherlands.

Scholten, H., Udink ten Cate, A.J., 1995. Testing simulation modelling quality, Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium, pp. II.A.3.1-6.

Scholten, H., Udink ten Cate, A.J., 1996. Quality assessment of simulation modelling. In: Lokhorst, C., Udink ten Cate, A.J., Dijkhuizen, A.A. (Eds.), Information and Communication Technology applications in Agriculture: state of the art and future perspectives. ICCTA, Wageningen, pp. 353-358.

Scholten, H., Udink ten Cate, A.J., 1999. Quality assessment of the simulation modeling process. Computers and Electronics in Agriculture 22, 199-208.

Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.

Scholten, H., Van der Tol, M.W.M., 1998. Quantitative validation of deterministic models: when is a model acceptable? In: Obaidat, M.S., Davoli, F., DeMarinis, D. (Eds.), The proceedings of the Summer Computer Simulation Conference. SCS, The Society for Computer Simulation International, San Diego, CA,USA, pp. 404-409.

Scholten, H., Van Waveren, R.H., Groot, S., Van Geer, F., Wösten, H., Koeze, R.D., Noort, J.J., 2000. Good Modelling Practice in water management, Proceedings HydroInformatics2000. International Association for Hydraulic Research, Cedar Rapids, Iowa, USA.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze, J. Noort (1999). Good Modelling Practice Handbook, STOWA, Utrecht, RWS-RIZA, Lelystad, The Netherlands (in Dutch, English version from http://www.landandwater.tudelft.nl/downloads/GMP-uk.pdf).

Walker, W., P. Harremoes, J. Rotmans, J. Van der Sluijs, M. van Asselt, P. Janssen, and K. v. K. MP. 2003. Defining uncertainty. A conceptual basis for uncertainty management in model-based decision support. Integrated Assessment 4:5-17.

Wigan, M.R., 1972. The fitting, calibration and validation of simulation models. Simulation 18, 188-192.

Young, P.C., 1983. The validity and credibility of models for badly defined systems. In: Beck, M.B., Van Straten, G. (Eds.), Uncertainty and forecasting of water quality models. Springer-Verlag, Berlin, Heidelberg, New York, pp. 69-98.

2 Methodological building blocks

"The time has come," the walrus said, "to talk of many things:
Of shoes and ships - and sealing wax - of cabbages and kings"
(Lewis Carroll, pseudonym of Charles Lutwidge Dodgson, mathematician and logician, 1832-1898)

Content of Chapter 2

2.1 Setting the scene

The modelling problems and the research questions outlined in Chapter 1 will not be coped with from scratch. There exist a number of building blocks that will be used as the fundament, on top of which proposed solutions and answers will be erected. These building blocks consist of methods; theories, tools and vocabulary that are instrumental to the chosen approach and originate from a wide variety of disciplines.

This book is not focused on *decision support*[1] but on *problem solving*[2], which is of course also of paramount importance as a basis for decisions and their support. Model-based problem solving encompasses understanding a problem at hand, the desired situation, a solution-space with possible alternatives and methods to that bring the present (problematic situation) to a desired one. Model-based problem solving focuses on how a problem can be solved: in this case by using a model.

Wierzbicki *et al*. (2000) use the term model-based decision support and they state that the dominating aspects of decision-making are: (i) information about the current situation and history, (ii) the relation between basic processes and actions or decisions, and (iii) the decision process. In this book the first of these aspects (i) will be referred to as the body of knowledge on *problem and object system*. The second aspect (ii) will be discussed here as *mathematical model* (often abbreviated to model). The third aspect (iii) will be narrowed to *modelling*[3].

The following building blocks are used and will be briefly explained in the remainder of this chapter: *systems science* (section 2.2), *mathematical models* (section 2.3), *quality assurance* (section 2.4), *structured modelling & simulation* (section 2.5), *relational databases* (section 2.6), *knowledge based systems* and *artificial intelligence* (section 2.7), *process defining technologies* (section 2.8) and, *software engineering* (section 2.9). Section 2.10 summarizes how these building blocks are used throughout this book.

2.2 Systems Science

2.2.1 Introduction to systems science

The term *cybernetics* dates back to the 1940's and 1950's when thinkers such as Wiener, von Bertalanffy, Ashby and von Foerster founded cybernetics through a series of interdisciplinary meetings. Since the 1960s the closely related term *systems science* is in use. Systems science is the scientific approach to analyze natural systems and design and analyze technical and organizational systems. Systems science looks at things in terms of *systems*, *processes* and *feedback loops*. Systems science is a toolbox filled with practical applicable methods and techniques. It is a mix of several ingredients, e.g. systems thinking, modelling & simulation, design methodology, decision theory and decision-making. It touches several disciplines, including mathematics, computer science, engineering, technology, biology, philosophy, social science and business science. At present, systems science has influenced many disciplines and has extended their former classical theoretical/analytical approach. Without systems science models will not have sufficient representation power. Furthermore, systems science enables us to combine knowledge on problem domains and provides a pragmatic way to deal with restrictions and hurdles within a single scientific paradigm.

The term *system*, from the Greek *σύστημα*, can be defined[4] as an assemblage of *entities* or *objects*, real or abstract, comprising a whole with each and every component/element interacting or related to at least one other component/element. Any object without a relationship with any other element of the system, is not a component of that system. A *subsystem* is then a set of elements, which is a system itself, and a part of the whole system.

---

[1] Turban (1995) defines it more specifically as *an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making* (from Wikipedia, http://en.wikipedia.org/wiki/Decision_support_system).
[2] Britannica Concise Encyclopedia (http://www.answers.com/topic/problem-solving-3) defines problem-solving as: *process involved in finding a solution to a problem*
[3] Sometimes the much broader and generic term *process* will be used. *Modelling* can then be seen as *a process* or an instance of *process*.
[4] From Wikipedia: http://en.wikipedia.org/wiki/System.

Systems science can be regarded as one of the ancestors of the presently developing *sciences of complexity*, including AI, neural networks, dynamical systems, chaos, and complex adaptive systems. This development assumes that however complex or diverse the world that we experience, different types of organization will always be found and this organization can be described by concepts and principles, which are independent from a specific disciplinary view. Therefore, uncovering those general laws would permit to analyze and solve many different types of problems.

2.2.2 History of systems science

Since about 1950 there has been a parallel development in several disciplines (e.g. social sciences, psychology, biology, etc.) related to what later became system science. Wolfgang Köhler introduced the concept of physical *Gestalten*, culminating in the hypothesis of psychophysical isomorphism (Köhler (1924, 1927). Lotka introduced the concept of open systems (Lotka, 1925). The term model is much older, but some of the older mathematical models are still in use, e.g. Verhult's *logistic growth* (Verhulst, 1838), the Lotka-Volterra model of oscillating chemical reactions (Lotka, 1920) and the famous *Lotka-Volterra model* of predator prey interaction (Lotka, 1925, Volterra, 1926). The biologist Karl Ludwig von Bertalanffy introduced an individual growth model (Von Bertalanffy, 1934) and later the General System Theory (Von Bertalanffy, 1968). A list of founders of systems science should also include the following persons: Walter Bradford Cannon who introduced *homeostasis* (Cannon, 1932), the Chilean biologists Humberto Maturana and Francisco Varela who introduced *autopoietic systems* (Maturana and Varela, 1973), the theoretical biologist Robert Rosen who introduced *anticipatory systems* (Rosen, 1985), the economist Boulding who related human economic behavior with other behavior in a larger interconnected system leading to what he called the General Systems Theory (Boulding, 1956), Miller as creator of the *living systems theory*, which aimed to formalize the concept of life (Miller, 1978), Stafford Beer, best be considered as a core theorist in the field of management cybernetics, involved in Operations Research, as creator of the *Viable System Model* (Stafford Beer, 1972), the systems scientists, Peter Checkland, who introduced *soft systems* (Checkland, 1981 and Checkland, and Scholes, 1990), and A. Wayne Wymore, as creator of the *Tricotyledon Theory of System Design* (T3SD), which is a mathematical theory of systems engineering developed by him (Wymore, 1967, 1976).

Parallel to this development contributions have been brought to systems science by scientists working in *cybernetics*. The theoretical and applied mathematician Nobert Wiener was a pioneer in the study of *stochastic and noise processes*. His work is used in electronic engineering, electronic communication and control systems (Wiener, 1948).

Jay Wright Forrester is the inventor of *System Dynamics*, which deals with the simulation of interactions between objects in dynamic systems (Forrester, 1961). He was also the mind behind the World2 model, the ancestor of the World3 model, developed by his protégé Donella Meadows in cooperation with Dennis Meadows, and later Jørgen Randers (Meadows *et al*., 1974). This was the model that formed the basis for the Club of Rome study *Limits to growth* (Meadows *et al*., 1972).

A more formal approach is chosen in systems theory, which is a transdisciplinary / multiperspectual theory that studies structure and properties of systems in terms of relationships from which new properties of wholes emerge. In 1954 Von Bertalanffy, Boulding, Rapoport, and Gerard founded the *Society for the Advancement of General Systems Theory*, (called *Society for General Systems Research* in 1955, renamed the *International Society for General Systems Research* in 1986, called the *International Society for the Systems Sciences* since 1988.

Some scientists have disseminated advances of system science to other paradigms. A good example of these is Russell Ackoff, architect by profession, who bridged the gap between Operations Research and social sciences and imported systems science into Operation Research (Ackoff, 1960). The famous Herbert Simon, winner of the Nobel Prize in economics, disseminated many concepts of systems science to computer science, management science and many other disciplines (Simon, 1957, 1962).

2.2.3 A systems approach

A systems approach, i.e. using systems science, distinguishes itself from a more conventional and analytical approach by focusing on the whole system, its components and interactions more than on reducing a system to its elementary elements in order to study in detail and understand the types of interaction that exist between them. In this way both the systems approach and the analytical approach are complementary, especially in case of complex systems[5].

Herbert Simon distinguishes *natural sciences*, e.g. biology, physics, chemistry, and *artificial sciences*, e.g. cybernetics, cognitive sciences, decision science, organization science (Simon, 1996). Opposite to the objects of natural sciences (e.g. chemical processes, organs, cells, crystals), the objects of the artificial sciences (e.g. information, decision, communication, concepts) do not match with the classical epistemology[6] and its experimental methods and refutability[7]. It is the task of natural sciences to comprehend, while the artificial sciences focus on engineering and design of the artefacts[8]. But just as natural objects, artefacts have to obey natural laws such as gravity. Science is focused on *analysis*; engineering is focused on *synthesis* and design. Engineering is concerned with designing artefacts with a purpose and a function. Although the artificial sciences cannot be treated as natural sciences, many of the methods from natural sciences can be used, but the artefacts are often complex and analytical methods from natural sciences are not sufficient, as understanding is not the only focus, but proper functioning. Here a systems approach may contribute, as it is not focused on understanding all details, but on the interactions between the artefact and its surroundings (e.g. its user).

2.2.4 Systems Science as building block

In this book a *systems approach* is used to view different aspects of model-based problem solving, including the problem and the (object) system, in which the problem is embedded and the model that reflects some essential aspects of the *object system*[9]. Typically the object system contains components that are related to each other. In open systems there are also relations between system components and entities in the environment of the system (Figure 2-1).

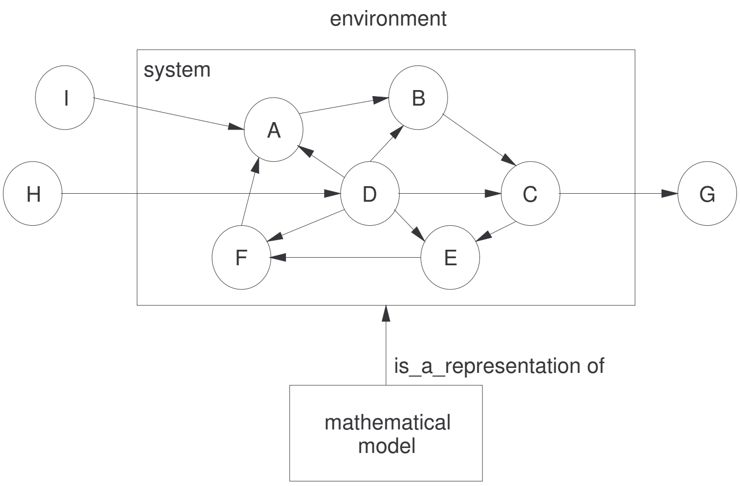


Figure 2-1. The object system (OS) of a problem described in a system science sense and a model based on this systems science view. *Circles* are entities and *arrows* represent relations. Circles within the problem/OS rectangle are included in the system, while the other circles are part of the environment of the system.

[5] Herbert Simon gives a rather informal definition of complex systems: *[…] by a complex system I mean one made up of a large number of parts that have many interactions. […] in those systems the whole is more than the sum of the parts in the weak but important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole.*

[6] Wikipedia: http://en.wikipedia.org/wiki/Epistemology: *Epistemology*, from the Greek words επιστεμε (*episteme*, knowledge) and λογοσ (*logos*, word/speech) is the branch of philosophy that deals with the nature, origin and scope of knowledge.

[7] WordNet 2.1 (3rd meaning): Refutation is the act of determining that something is false [synonym: *falsification*].

[8] *Artefact* means here human-made products of the artificial sciences.

[9] In Chapter 5 *object system* will be defined as *the things and processes in the real world or some defined system that will be the object of modelling*.

Systems science basically assumes that although many problems may not be similar, many can be solved along analogous lines. The systems, to which the problems belong, are similar if viewed from the right angle. This permits reusing (parts) of the problem-solving methodology.

2.3 Mathematical models

2.3.1 Introduction to mathematical models

'Modelling' refers to developing and/or using models in order to solve some problem, in other words it can be described as *model-based problem solving*. These kinds of models can further be categorized as *mathematical models*, as they share the use of mathematics to represent relevant parts of an object system and to solve the problem at hand. Mathematical models are abstractions of (some) aspects of the real world in a mathematical form typically consisting of one or more equations. Models consist of a *model definition* (often represented in mathematical format), a method to solve the model, the *model solver* and *data*, needed to solve the model. The model definition (format) determines the type of a model and the model solver that can handle that type of model.

Mathematical models can be classified in many ways. One way is looking at what kind of knowledge is incorporated in a model. If the knowledge consists of scientific (state-of-the-art) knowledge on one or more processes in a mathematical form (equations/inequalities), it is often referred to as *process* (based) *model*, opposed to an *empirical model*, which is often based on observations of the real system (some expected statistical relationship). The process-based models aim at understanding, while the empirical models focus on prediction. Mixes of these two types combine available process knowledge with empirical model components to fill process knowledge gaps or for efficiency reasons[10].

Some other ways to classify mathematical models are *static* versus *dynamic*, *linear* versus *non-linear*, *lumped parameter model* versus *distributed parameter models*, *deterministic* versus *stochastic*, etc. Here we focus on two types of mathematical models: simulation models (see section 2.3.2) and models used in Operations Research (see section 2.3.3). Both types can also be classified according to the above mentioned categories.

Simulation models aim at simulating the object system in terms of behavior or in terms of the structure. In the first case, the outcomes of the simulation model should look like direct or indirect observations of the real world, while in the second case the model should also mimic the dominant components and the associated processes of those aspects of the real world that are included in the definition of the object system and therefore indirectly of the real world.

Models used in Operations Research are also mathematical models that represent some chosen aspects of an object system, which these models do not aim to reach a predefined goal or to reproduce system behavior.

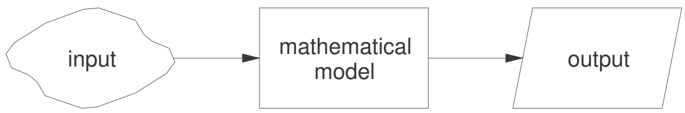


Figure 2-2. A systems science view on modelling (Elzas, 1980).

Depending on the reason of using a model in model-based problem solving, one is interested in the following (Figure 2-2):

- If the input and the model are known, the object is *prediction* and the model is typically a simulation model;
- If the input and the output are known, the object is model *identification / calibration*[11] (structure, parameters);

[10] Models completely based on process knowledge are often called *white-box-models*; empirical models are also known as *black-box-models* and models that combine both elements are referred to as *grey-box-model*.
[11] The terms *identification* and *calibration* will be explained in section 2.3.3.

- If the model and the output are known, the objective is to find the specific input that, assuming that the model is 'true', produces the given output. This is the case in management or in using the model as an operational decision instrument, typically what *optimization* in Operations Research is focused on.

In reality things are not so simple, as there are many complications. To a large extent these can be attributed to uncertainty, because models are based on insufficient theoretical and experimental scientific knowledge on dominant processes governing those aspects of an object system that are relevant for the problem at hand. This is often true in softer domains, i.e. ecological, biological, environmental, socio-economical models and in compound models, covering hard, technical domains, i.e. hydrology, as well as one ore more of the softer domains. Many other sources of uncertainty of model outcomes worsen these model credibility problems. In many cases of real world problems a lack of system observations hamper a proper model calibration. In addition to these credibility problems, many other shortcomings can be noticed. Sometimes these are related to model content and lack of data, i.e. using the 'wrong' model, sometimes the problems are caused by an undefined and unclear modelling process. The model content and system observation related shortcomings result in large uncertainties. O'Neill and Gardener (1979) discussed the sources for uncertainty in the outcomes of ecosystem models: (1) model bias or errors in model structure, (2) measurement error or uncertainty in model parameters and (3) variability of natural ecosystems. Furthermore, shortcomings in the modelling process result in irreproducible modelling projects, impeding modelling playing a sound role in decision making.

In general the pitfalls of model uncertainty and sloppy modelling arise from the overall characteristics of modelling. Models are simplifications of reality. A useful definition of modelling is given by Minsky (1965):

> *To an observer* B*, an object* A^* *is a model of an object* A *to the extent that* B *can use* A^* *to answer questions that interest him about* A.

In order to be useful in the context of this book, this definition has to be extended to include more terminology. Instead of just extending the definition, a model will be described in the context of modelling itself, in a mental process model[12] on model-based problem solving.

Table 2-1. First draft of a *mental process model* outlining some basic terminology related to mathematical modelling and model-based problem-solving. This *mental process model* will be improved and extended in Chapter 6 and Chapter 7.

Mental process model on model-based problem-solving, version 0

1. A *problem owner* has a (real world) *problem* to be solved.
2. The *problem owner* finds a modeller or *modelling team*.
3. The *modelling team* specifies an *object system*, which should contain all aspects of the real world that are relevant for the *problem* at hand[13].
4. *Problem owner* and *modelling team* should discuss and agree on the *object system*.
5. *Problem owner* and *modelling team* should define, discuss and agree on criteria for model evaluation.
6. The *modelling team* selects an appropriate existing model or develops a new one by formalizing the *object system* in a (mathematical) *model*.
7. The *modelling team* adds (input) data from the *object system* to the (mathematical) *model*.
8. The *modelling team* analyzes the model (qualitatively and quantitatively according to predefined criteria (step 5)).
9. The *modelling team* uses the *model* to solve the (real world) problem at hand.

An overall diagram of this model-based problem solving definition is given in Figure 2-3. There are various actions[14] that can be included in this process, e.g. comparing the observations with the output. Depending on

[12] From Wikipedia (http://en.wikipedia.org/wiki/Mental_model): A *mental model* is an explanation in someone's thought process for how something works in the real world.
[13] In case of mathematical simulation models, the object system should contain the (relevant) structure and relevant process(es).
[14] Action is used here as a fuzzy, undefined concept, referring to things to do in model-based problem-solving. Later this fuzzy concept will be refined and made explicit.

the reason why this is done and how this is done, such a comparison of input and output can be seen as model (performance) *evaluation*, *validation* or *calibration*.

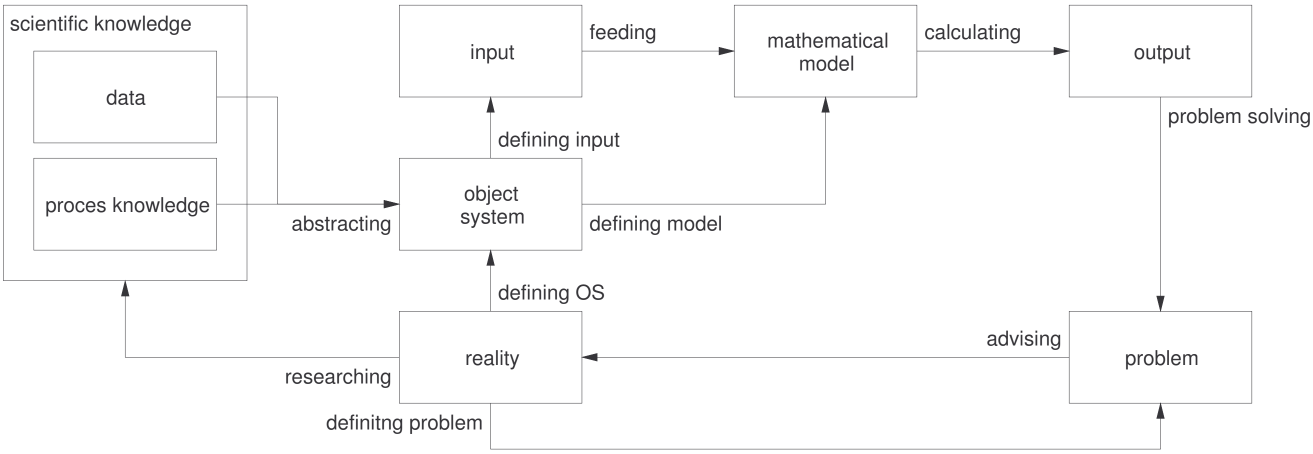


Figure 2-3. A new view on model-based problem solving, illustrating the definition of model-based problem solving. *Rectangles* represent entities and *arrows* represent actions.

In the case of *model evaluation*, the difference between model output and system observations is used as a measure on how well a model predicts an observed system behavior. *Model calibration* uses differences between output and observations to adapt the model within the boundaries, provided by the process knowledge incorporated in the model. *Model validation* assumes that the model is evaluated for a different (but comparable) system, i.e. with other input. Besides these procedures in model-based problem solving, there are many more, dealing with problem formulation, data handling, model implementation, sensitivity analysis and interaction between modeller and problem-owner. The large number of *things to do*, the lack of agreement on how these fit in an overall scheme and because such a scheme will help stakeholders in model-based problem-solving, in section 2.5 an overview will be given on structured modelling and simulation.

2.3.2 Mathematical models for *modelling and simulation*

Despite the fact that the early days of *modelling and simulation* belong to the distant past, its practical use and popularity is strongly coupled to the era of computers, which enabled scientists to solve large numbers of algebraic equations or sets of differential equations automatically. The latter is essential to be able to develop and use complex simulation models. In the 1970s a series of more systematic and profound studies aimed at providing a theoretical foundation for modelling. Three names are strongly associated with this theoretical foundation for simulation modelling: Bernard P. Zeigler, Tuncer I. Ören and Maurice S. Elzas. Their series of books (Zeigler, 1976, 1984, Zeigler *et al*. 1979, Ören, 1978, Ören *et al*, 1984, Elzas *et al*., 1986, 1989) was the stage for publication of novel views by many authors whose contributions to the modelling process and its technology are still very relevant today. Other key-persons to be mentioned here are: Klir (for his General Systems Problem Solver, (Klir, 1985). Next to the 'godfathers' of *modelling and simulation*, some of these authors inspired me and have to be mentioned here including *Sargent* for a theoretical view on verification and validation (Sargent, 1982, 1984a, 1984b), *Goldberg* and *Davis* for using Genetic Algorithms to solve optimization problems in simulation modelling (Davis, 1987, Davis and Coombs, 1989, Goldberg, 1989, 1994), Cees de Wit as founding father of production ecology models (de Wit, 1960, 1965), and finally *O'Neill* and *Gardner* on uncertainty in model outcomes (O'Neill and Gardner, 1979, O'Neill *et al*., 1980, Gardner *et al*., 1981, Gardner *et al*., 1982, O'Neill *et al*., 1982a,b, Gardner and O'Neill, 1983).

Elzas attempted to bring together the paradigms of systems science and that of modelling and simulation on one hand and to unite different views on model development on the other hand in a seminal paper titled 'System paradigms as reality mappings' (Elzas, 1984). Furthermore, he discussed various model development approaches. His distinction between a phenomenological approach (modelling based on raw system observations), an inductive approach (modelling based on *a priori* knowledge of the system at hand) and a deductive approach (modelling based on *a priori* knowledge but only insofar as it is supported by observational evidence) are still valuable styles in model development (Elzas, 1984).

2.3.3 *Modelling and simulation* models for ill-defined systems

Many mathematical models are dynamic, continuous non-linear simulation models defined in differential and algebraic equations. The problems and object systems for which they are developed typically belong to the domains of environmental studies, ecology, social science or managerial science. In that case, one often speaks of models for ill-defined problems (Van Straten, 1986, 1998, Beck, 1987), as the observational data and scientific prior knowledge do not allow defining a single-best model, but a series of alternative models that are equally able to explain the observations (i.e. field measurements, data), given the included knowledge.

A dynamic model aims at describing the structure and behavior in time of some defined object system using so called *state variables*. Ideally, these state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time. Such models typically consist of differential equations and can be represented in a state space form (see Appendix A).

A mathematical model for *modelling and simulation* needs to be solved. If such model is a continuous simulation model consisting of a set of ordinary differential equations, many *solvers* exist. Typically modelling software and tools[15] include some solver(s) and, furthermore, many methods are available in relevant handbooks (e.g. Press *et al*., 1992).

Several components of model equations permit to deal with various kinds of uncertainty. The measured input variables for instance reflect natural variability, which hinders its use for similar but other situations (prediction, extrapolation). *Decision variables*[16] can be used to identify which scenario or man made decision will cause the desired future situation, according to the model. Model parameter values are derived from previous research, but (1) some cannot be measured precisely or only indirectly in some lab experiment, (2) some are accurately known (constants) and (3) some do not have much impact on the model outcomes. Parameters of category (1) have to be identified in model calibration, while those of group (2) and (3) can be kept constant during calibration[17]. Performing a *sensitivity analysis* can facilitate the classification as to which category a parameter belongs.

Many activities are required in a proper modelling practice, including model development, model analysis and model use. How to handle the complex scheme of what to do is the subject of section 2.5 and will be discussed in detail in Chapter 9. Here only model calibration/identification and model validation will be briefly discussed, as this touches the building block on Operation Research, discussed in section 2.3.4.

Model identification aims at identifying if a model contains the right structural components and processes. This also means finding optimal values of uncertain parameters, initial values of state variables and uncertain inputs (including decision variables). These optimal values and a unique best model should give perfect model output, given (historical) system observations. Models developed for ill-defined problems can therefore fail. In that case one can *calibrate* the model to find a certain degree of fit within the range of predefined accuracy between observations and model outcomes, leading to a reduction in model output uncertainty. For *model identification* and *model calibration* the same set of observations can be used. *Validation* is substantiation that a model, within its domain of applicability, possesses a satisfactory range of accuracy consistent with the intended application of the model (Schlesinger (1979). To validate a model one needs at least one new, independent set of observations.

A model is too complex or over-determined model, if it has more state variables and parameters, i.e. the model's structure is too complex, than can be determined with available system observations (i.e. measured data) in an identification procedure. This refers to Beck's dilemma, i.e. the difficult choice of a modeller between a simple model, which can be identified by system observations completely, and a more comprehensive model, which incorporates all available knowledge (Beck, 1987, Scholten and Smaal, 1999). The simpler model would be capable of 'predicting' past system behavior correctly, but it would not be able

[15] A rather complete overview of Andrea Emilio Rizzoli http://www.idsia.ch/%7Eandrea/simtools.html.
[16] A *decision variable* or *control variable* can be defined as: input variable used to calculate the effects of (human) intervention in the system with the model.
[17] Depending on the aim of the model study, such parameters and associated processes can also left out of the model, unless the model aims to include all detailed scientific knowledge.

to predict the future under substantially different conditions. The comprehensive model would intuitively be supposed to be superior, but its apparent redundancy would decrease our confidence, because it could generate equally probable, but quite contradictory predictions (Beck, 1978). This leaves us with Beck's dilemma: a simple model predicting 'wrongly' but in a highly precise fashion or a comprehensive model predicting 'correctly' but is highly imprecise (Scholten and Smaal, 1999). This problem will not be solved in the context of this book.

2.3.4 Mathematical models for Operation Research (OR)

Operations Research[18] or simply OR can be defined[19] as an interdisciplinary science which uses mathematical modelling, statistics, and algorithms for decision making in complex real-world problems. OR aims at eliciting a best possible solution to a problem. The terms operations research and management science are often used synonymously, but management science generally implies a closer relationship to the problems of business management.

The methods used in OR include statistics, optimization, stochastic approaches, queuing theory, game theory, graph theory, simulation and computer science, of which optimization is the most important. *Optimization*, sometimes called *mathematical programming*[20] refers to a scientific paradigm, which aims at minimizing or maximizing some mathematical simple or very complex function by systematically varying its arguments (typically decision variables) within some *constraints* or *criteria*. The aim is to find values of the decision variables that give rise to a feasible solution. A feasible solution that optimizes (i.e. minimizes or maximizes) the object function is the optimal solution. The function itself is often called the *objective function* or *cost function*. A general format is presented in Appendix A.

2.3.5 Mathematical models as building blocks

The mathematical models discussed in this book, i.e. those typically consisting of sets of differential equations, belong to the type described in section 2.3.3. The models aim at understanding ill-defined systems, originating mainly from environmental and ecological domains. These models are hard to identify, which only allows calibrating the model. Calibration aims at finding one or more parameter vectors and vectors of initial values of the state variables that realize a satisfactory fit between model outcomes and observations in the real system, in turn also reducing model output uncertainty.

The approach chosen for continuous modelling and simulation models for environmental and ecological domains in this book, combines the OR-approach (optimization of one or more object functions within a series of constraints) with the mathematical approach (see Appendix A, section 2.3). In this way the simulation model is used as a complex function and the objective function will be defined based on the distance between the model outcomes and the associated system observations. If this distance can be expressed in a single measure, i.e. the object function, calibration is an optimization problem, aiming at minimization of the object function by varying the uncertain model inputs, i.e. the uncertain parameters, the uncertain initial values of the state variables and decision variables.

If a simulation model is calibrated in this way and the model appears unidentifiable, the residual distance between model outcomes and system observations is larger than zero, reflecting the *a posteriori* model uncertainty, which should not be large or systematic. If so, the model can be used within its scope, which is partly determined by the conditions provided by the calibration setting, e.g. the knowledge on the real system as reflected by observations.

---

[18] The term Operation Research originated in World War II, when (mainly British) mathematicians looked for better ways to make decisions on various military problems, e.g. the optimal size and speed of convoys of ships crossing the Atlantic Ocean with supplies.
[19] Adapted from Wikipedia: http://en.wikipedia.org/wiki/Operations_research.
[20] Programming here does not refer to writing computer programs, but originates from the US military and refers to a program for training and scheduling.

2.4 Quality Assurance

2.4.1 Introduction to Quality

In his benchmark book on quality, Crosby (1979) uses sex as a metaphor for quality:

> *"The problem of quality management is not what people don't know about it. The problem is what they think they do know….*
> *In this regard, quality has much in common with sex. Everybody is for it. (Under certain conditions, of course.) Everyone feels they understand it. (Even though they wouldn't want to explain it.) Everyone thinks execution is only a matter of following natural inclinations. (After all, we do get along somehow.) And, of course, most people feel that problems in these areas are caused by other people. (If only they would take the time to do things right.)."*

There are many reasons to adopt quality assessment as an integral part of all model-based problem-solving studies. Quality is not the result of 'following natural inclinations', but a meta-process which demands a lot of self-control and hard work (Scholten and Udink ten Cate, 1999).

Measuring quality is part of everyday life. By partaking food one can evaluate its taste, its freshness, its structure, and its smell. Judges in sport disciplines like ice dancing, horse dressage or platform diving translate aspects such as complexity, artistic value and the harmony of a performance into a simple report mark. All of these quality factors are subjective and even when quality is measured using some kind of metrics, it will only rarely be objective: arbitrary and subjective elements cannot be excluded. Here the main topic is quality improvement instead of quality evaluation (Scholten and Udink ten Cate, 1999).

In contrast to software development, quality assurance in modelling and simulation is not common practice. If scientists build such models, they see a model as a research tool and hardly as a product that can be used to support decision-making, scenario analysis or policy making. Usually modellers develop a model without considering the possibility that others may apply it. In domains of a more technical nature there is a tendency to improve and evaluate model quality. In domains with less hard, grey, models, which are based on incomplete, controversial theories and hypotheses, quality assessment is not a modeller's every day routine (Scholten and Udink ten Cate, 1999).

In section 2.5 quality aspects such as *correctness*, *reliability* and *usability* will be discussed with regard to the process of modelling and simulation and in Chapter 10 the overall framework, proposed in this book to improve the quality of model-based modelling will be discussed. Quality assessment in simulation is a very complex matter and includes different subjects such as quality assurance in software engineering, project management and several model validation items.

2.4.2 Quality assurance and ISO

2.4.2.1 The International Organization for Standardization (ISO)

ISO or the International Organization for Standardization is the world's leading standardization organization. ISO standards specify requirements for products, services, processes, materials and systems and for good management practice. ISO's members are the standardization and normalization organizations in most countries in the world. Besides ISO there are other international standardization organizations, e.g. CEN (Committee Européen de Normalisation or European Standardization Organization), IEC (International Electrotechnical Commission) and the ITU-T (International Telecommunication Union).

ISO has a current portfolio of more than 16000 standards that provide practical solutions and achieve benefits for almost every sector of business, industry and technology. ISO's standards do not only cover agriculture, construction, mechanical engineering, manufacturing, distribution, transport, medical devices, and information and communication technology, but also services.

2.4.2.2 ISO standards history

Errors committed in manufacturing have been an impetus for creating quality standards. For example, in World War I, a high percentage of shells failed to explode. This was traced to different definitions of an inch by the two major UK armaments manufacturers, leading to calibration standards. In World War II, the

United Kingdom had serious problems with accidental detonation in weapons factories and other manufacturing flaws. In an attempt to solve the problem, the Ministry of Defence placed inspectors in the factories to oversee the production process. To supply to the Government, a company had to write up the procedure used for making its product, have the procedure approved by the Ministry and ensure that their workers followed the approved procedure. Parallel to this, the United States developed requirements for contractors for shell, aircraft and missile suppliers during and shortly after World War II. In 1947 ISO was founded as an organization to develop and disseminate worldwide industrial and commercial standards. In 1959, the United States established Quality Program Requirements, MILQ-9858, a quality standard for military procurement, detailing what suppliers had to do to achieve conformance. By 1962, NASA had similarly developed Quality System Requirements for its suppliers. In 1963 MILQ-9858 was slightly improved (MILQ-9858 A). In 1968, NATO adopted the AQAP (Allied Quality Assurance Publication) specifications for the procurement of NATO equipment[21]. In 1979 British Standard Institute adopted these NATO standards and adjusted it for civilian use, i.e. BS 5750.

In 1987 ISO published an adapted BS 5750 as ISO 9000. New releases of ISO 9000 have been published since 1987 (e.g. 1994, 2000). ISO 9000 currently includes three quality standards: ISO 9000:2005, ISO 9001:2000, and ISO 9004:2000. ISO 9001:2000 presents requirements, while ISO 9000:2005 and ISO 9004:2000 present guidelines. In 2008 a new version of ISO 9000 will be published. All of these are process standards and do not apply to products.

2.4.2.3 ISO families of standards

The ISO 9000 and ISO 14000 families[22] of standards are widely known. ISO 9000 has become an international reference for quality requirements in business-to-business dealings and ISO 14000 helps organizations to meet their environmental problems. ISO 9000 and ISO 14000 families are known as *generic management system standards*. *Generic* means, that they can be applied to all kinds of organizations, for many types of products (even if a product is actually a service), in many sectors, and for different types of organizations, including business enterprises, public administrations, government departments and research institutes. *Management system* refers to what the organization does to manage its processes, or activities.

In the next sections ISO 9000 will briefly discussed (section 2.4.2.4) and its merits for modelling evaluated (section 2.4.2.5).

2.4.2.4 ISO 9000 quality standards in general

The ISO 9000 family of standards is initiated and maintained by ISO’s TC176 (Technical Committee on *quality management and quality assurance*) and is administered by accreditation and certification bodies. ISO 9000 includes the following standards[23]:

- *ISO 9000:2005, Quality management systems - Fundamentals and vocabulary*: covers the basics of what quality management systems are and also contains the core language of the ISO 9000 series of standards.
- *ISO 9001:2000 Quality management systems – Requirements*: is intended for use in any organization which designs, develops, manufactures, installs and/or services any product or provides any form of service. It provides a number of requirements, which an organization needs to fulfill if it is to achieve customer satisfaction through consistent products and services, which meet customer expectations. This is the only implementation for which third-party auditors may grant certifications.
- *ISO 9004:2000 Quality management systems - Guidelines for performance improvements*: covers continual improvement. This gives you advice on what you could do to enhance a mature system. This standard very specifically states that it is not intended as a guide to implementation.

[21] AQAP-1: *NATO Requirements for an Industrial Quality Control System* and AQAP-2: *Guide for the Evacuation of a Contractor’s Quality Control System for Compliance with AQAP-1*.
[22] See http://iso.nocrew.org/iso/en/aboutiso/introduction/index.html.
[23] Wikipedia: http://en.wikipedia.org/wiki/ISO_9000.

ISO 9001:2000[24] includes[25]:

- a set of procedures that cover all key processes in the business;
- monitoring processes to ensure they are effective;
- keeping adequate records;
- checking output for defects, with appropriate corrective action where necessary;
- regularly reviewing individual processes and the quality system itself for effectiveness;
- facilitating continual improvement.

Although the ISO 9000 way of working, i.e. defining and following guidelines from process definitions, monitor processes and keeping records and checking products for defects, will lead to some level of quality of processes and their resulting products, the basic idea is that the ISO 9000 is rather abstract and universal, i.e. applicable for a too wide a range of companies and organizations. Therefore, various industry sectors have developed their own interpretations of the ISO 9000 standards, on top the ISO 9000, with more specific requirements and more appropriate expert guidance. Specific guidance has been developed for a range of sectors (e.g. AS 9000 for aerospace manufacturers, PS 9000 for pharmaceutical packaging materials, QS 9000 for automotive manufacturers, TL 9000 for the Telecom Industry, TickIT for software development in UK, ISO/IEC 25000:2005 for Software product Quality Requirements and Evaluation (SQuaRE), ISO 13485:2003 for the medical industry's equivalent). These specific guidance/standards are therefore a particularization of ISO 9000 standards.

2.4.2.5 Are ISO 9000 quality standards sufficient for modelling?

The available quality standard, ISO 9000 and those built on top of the ISO 9000 family, are not focused on modelling and simulation. They are therefore not sufficient for modelling and simulation, because they are not specific enough (ISO 9000) or focused on different fields of application. The approach followed in the QA guidance document for model based water management matches with the basic ideas of ISO 9000, i.e. describing what to do (process definition), monitoring and recording what is done, comparing what is done with what should have be done and testing for satisfactory (intermediate) products and results (assessing the soundness of the conceptual model, calibrated model, validation and uncertainty analysis of scenario runs). All these are, to some extent, part of the modelling process and at the same time aiming at finding flaws in resulting products of the modelling process. The approach followed here combines testing products, e.g. model validation with a process approach.

ISO 9000 is concerned with *quality management*, i.e. what organizations do to enhance customer satisfaction by meeting customer and other relevant requirements and to improve their performance in this regard. ISO 9000 defines Quality Assurance as *'part of quality management focused on providing confidence that (the customer's) quality requirements will be fulfilled'*. ISO 9000 '*does not establish requirements for products*', as these '*can be specified by customers or by (an) organization in anticipation of customer requirements, or by regulation. The requirements for products and in some cases associated processes can be contained in, for example, technical specifications, product standards, process standards, contractual agreements and regulatory requirements*'.

Therefore, extra guidance for model-based water management, which is more specific, detailed and includes state-of-the-art expertise, fits perfectly in ISO 9000's Quality Assurance definition.

2.4.3 Quality aspects

Well-known aspects of quality, such as those defined by McCall *et al*. (1977) for software engineering, are also relevant to modelling and simulation (Table 2-2).

[24] ISO 9001:2000 includes the previous standards ISO 9002 and ISO 9003.
[25] Wikipedia: http://en.wikipedia.org/wiki/ISO_9000.

Table 2-2. Software engineering quality aspects in modelling and simulation, from Scholten and Udink ten Cate (1999), based on quality aspects for software engineering by McCall *et al.* (1977).

Quality aspect	*Meaning in simulation modelling*
Correctness	*Extent to which a model meets its specifications*
Reliability	*Extent to which a model can be expected to perform its intended function with a required precision*
Efficiency	*Amount of computing resources and code required by a model to perform its function*
Integrity[26]	*Extent to which access to model and data by unauthorized persons can be controlled*
Usability	*Effort required to learn, operate, prepare input, and interpret output of a model*
Maintainability	*Effort required to locate and fix an error in a model*
Testability	*Effort required to test a model to ensure it performs its intended function*
Flexibility	*Effort required to modify an operational model*
Portability	*Effort required to transfer a model from one hardware and/or software system platform to another*
Reusability	*Extent to which a model or parts of a model can be reused in other applications related to the packaging and scope of the functions the model performs*
Interoperability	*Effort required to couple one model to another*

2.4.4 Model validation

Apart from these aspects, borrowed from software engineering, model validation is the main quality aspect in simulation modelling. Model validation is discussed extensively in literature, but most authors offer merely a terminology instead of a methodology (Sargent, 1982; Sargent 1984a; Sargent, 1984b; Knepell & Arangno, 1993; Sheng *et al*., 1993, Oreskes *et al*., 1994, Rykiel, 1996, Refsgaard and Henriksen, 2004). Furthermore it has to be concluded that many models are to some extent *invalid*. This is especially true for models with a weak theoretical base. Validation alone will not improve model quality. Validated models reproduce system behavior correctly, but this does not imply that other quality requirements are met. The computer instance of that model may be hard to use or not flexible enough.

2.4.5 Quality assurance by meta-modelling

Humphrey (1989) defines a Capability Maturity Model (CMM), which classifies software producing organizations based on their operational practice belonging to one of the following stages: *ad hoc*, *repeatable*, *defined*, *managed* and *optimized*. According to Humphrey's model, Scholten and Udink ten Cate (1995) proposed a Simulation Maturity Model for organizations and projects with corresponding stages (Figure 2-4).

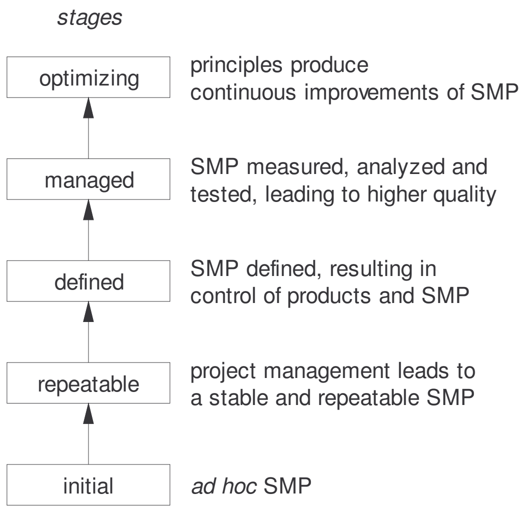


Figure 2-4. Stages in the Simulation Maturity Model (SMM), which aims at improving the simulation modelling process (SMP), from Scholten and Udink ten Cate, 1995).

[26] At present, I see *model integrity* in a wider meaning, including but not restricted to completeness, correct versions of model modules / submodels, correct input and model analysis data, model experiments within the scope of the model, etc.

2.4.6 QA as building block

Several concepts from this section 2.4 are used in this book to improve model-based problem solving. These include the general ideas behind the ISO-9000 standards, described in section 2.4.2, but here guidelines will be developed that are more specific for modelling and simulation. In addition, the Simulation Maturity Model of section 2.4.5 will be adopted and used at two of its stages: the *defined stage* and the *managed stage*. A complete description of the process of modelling and simulation will be developed to reach the *defined stage*. This description should be as formal as possible. Furthermore a tool will be developed to support modellers and others involved in their modelling work to tackle the *managed stage*. Finally, some ideas are formulated on how to use results of modelling projects, performed with the tool to learn from previous projects and reach what the *optimize stage* of the SMM aims for (Scholten *et al*., 2007).

2.5 Structured modelling and simulation

2.5.1 Introduction to structured modelling and simulation

As already set in sections 0 and 2.4, simulation modelling is a complex process or network of *things-to-do*. Since many years schemes have been proposed to recommend what has to be done and this book does a next attempt, aiming at a more formal approach. In this section a summary will be given of relatively recent previous efforts, under the header *structured modelling and simulation*[27]. It will include *ad hoc* modelling and simulation schemes and more focused good modelling practice guidance.

This section focuses on *modelling* and not on models or the problems at hand. The modelling process consists of interlinked activities of one or more persons. The historical perspective summary given here is mainly fuelled by my own experiences. The discussed items follow the order, in which they occurred, i.e. *ad hoc* modelling schemes, Good Modelling Practice, formal simulation modelling norms and the HarmoniQuA project.

2.5.2 *Ad hoc* modelling schemes

Many others contributed to a methodological framework for modelling and simulation (modelling and simulation), which belongs to the present expertise of most modellers. This methodological framework is valuable for the community of model based problem solvers as well, but it is fuzzy and non-explicit. Their modelling schemes are adapted by many authors and usually consist of networks of actions and / or products related to using models or to developing models. Typically the authors of a paper or book develop these informal schemes. At best the schemes are shared by a group of modellers.

Many other authors have proposed schemes to define the modelling and simulation process, of which the most interesting emphasized intended to test, verify and validate intermediate products of the modelling and simulation process (Sargent, 1982, 1984a, b, Knepell and Arangno, 1993). The main shortcomings of these approaches and those of Scholten (1994) and of Scholten and Udink ten Cate (1995, 1996, 1999) are associated with the lack of approval by a large group of modellers. An example Scholten and Udink ten Cate (1999) is shown in the Appendix A.

2.5.3 Good Modelling Practice

In 1997 several projects started to improve and facilitate modelling for water management in the Netherlands. One of these projects aimed at developing a Good Modelling Practice Handbook (Van Waveren *et al.*, 1999, Scholten, 1999, 2001, Scholten *et al.*, 2000, 2001, Blind *et al.*, 2000,

The foundation for the GMP Handbook is the assumption that defining the modelling and simulation process promotes its repeatability and facilitates an efficient and effective audit (see also section 2.4.5). The definition of the modelling and simulation process that was constructed on this basis has been used as the backbone of a Good Modelling Practice (GMP) handbook. The GMP-approach is comparable to the ISO-9xxx paradigm, but ISO standards are too rigid on one hand and too generic for model-based problem-solving.

[27] This should not be confused with Geoffrion's *structured modelling* as his models are focused on optimization and not on simulation (Geoffrion, 1987).

The efforts to define the modelling and simulation process for the GMP handbook focus on general agreement and negotiation, both from a user-participation point of view, as well as from a knowledge-acquisition point of view. The *ad hoc* modelling schemes of section 2.5.2 are usually not based on agreement within a group of expert modellers, as is the case for the GMP handbook. This main difference denotes the difference between knowledge as such and knowledge structured with the help of an ontology. The GMP ontology is not a formal one, but highly informal[28], achieved by decomposing the modelling and simulation process in steps. These steps have been decomposed further until a decomposition level has been reached, at which what has to be done is simple and straightforward for most modellers in water management. The informal ontology derived in this way was subsequently used as the core of a Good Modelling Practice handbook. The major payoff of developing such a handbook is its (informal) ontology of shared and approved 'simulation and modelling concepts'. Many modellers in the field of water management did not expect a final consensus on this complex and confusing matter.

The backbone of the handbook is the informal ontology. At the highest level of the decomposition the steps are: (1) starting with a logbook, (2) defining the modelling project, (3) building the model, (4) analyzing the model, (5) using the model, (6) interpreting the results, and (7) reporting and archiving (see Appendix A). These steps are further decomposed to a level of a single modelling activity (e.g. determining for which factors the model is most sensitive or checking if all model objectives are met). The core part of the handbook includes also a series of tests without prescribing with which methods or with which algorithms these tests have to be carried out. These tests comprise: conceptual model validation, some aspects of verification (dimension check, mass or energy balance control), a robustness test, a sensitivity analysis, a calibration, a (historical data) validation, and an uncertainty analysis. In this way the handbook supports all activities related to modelling and simulation and passing these tests improves the credibility of the model.

2.5.4 Formal simulation modelling norms

The GMP handbook discussed in section 2.5.3 has been received with enthusiasm, but as more or less anticipated the intended users of the GMP handbook were not eager to use it in their daily practice. In order to improve its use along two lines initiatives have been taken. The first consists of enforcing its use by the client of the modelling project. The second aimed at upgrade the status of the handbook by transforming it into formal national[29] norms.

At the initiative of the Institute of Inland Water Management and Waste Water / RIZA, the *Nederlands Normalisatie Instituut* (NEN) has upgraded the GMP handbook to a series of norm designs with two different levels of formality, 2 real norm designs for parts of the handbook with sufficient maturity and a design which consists of a series of practical advices. At present the whole set consists of:

- Norm Design[30] NEN 6260-1: *Termen en definities van modelleren* (in English: Terms and definitions of modelling)
- Norm Design NEN 6260-2: *Basisprincipes van modelleren* (in English: Basic principles of modelling)
- Practice Guideline Design NEN 6260-3: *Valkuilen gevoeligheden van modelleren* (in English: Pitfalls and sensitivities of modelling)

This set of norms is strongly based on the GMP handbook. Basically the text of the handbook has been used for these norms, but many errors, misconceptions, inconsistencies and fuzzily formulated concepts were improved. Especially the glossary, which is the major part of NEN 6260-1, is more consistent and provides a transparent view on the terminology used in the other parts of the norms.

The norms are a significant improvement compared to the GMP handbook. Its status is higher and its quality has been improved. The major disadvantage of the GMP handbook is shared by the set of norms. Both have a sequential approach, which focuses on reading it as a book. In the daily practice of modellers, familiar with

[28] Ontologies, as such and differences between *informal* and *formal* ontologies will be discussed in Chapter 3.
[29] *National* is *Dutch here*.
[30] *Design* refers here to the preliminary character of the norms and practice guideline. The procedure to make it formal is not completed yet, due the initiating institute (RIZA) and the normalization institute (Nederlands Normalisatie Instituut) on contract interpretations.

the norms or the GMP handbook, its use is not serial, but one typically follows the order, in which tasks have to be performed, starting with the fill-in forms. If the questions to be answered or the activities to specify in the fill-in forms are not clear, the modeller switches to the part of the norms (or the handbook), which defines these things. Unfamiliar terms can be clarified in the glossary and generic of domain specific pitfalls can be read and avoided in parallel with the task to perform. In this way users typically browse in a non-serial way through the norms (or the handbook). Just as for the GMP Handbook, this sequential character is the main disadvantage of the set of norms.

2.5.5 HarmoniQuA

Managers of problem-solving projects in water management have to arrive at a shared vision on the nature and extent of a modelling project, in which solutions have to be found to a stated management problem. Such a vision entails the scope of the study, the solution approach, expected results, duration, costs and resources used. Thereafter, for a commissioned project, the problem is to execute it within its specifications including quality assurance issues. In this way, transparency is guaranteed and projects are easier to audit and reconstruct (Scholten *et al*., 2007).

Quality assurance requirements for modelling projects are caused and fuelled by a multitude of problems and bad experiences with model based studies in the past. Refsgaard *et al*. (2005) and Scholten *et al*. (2007) give several reasons for these problems, including ambiguous terminology, a lack of mutual understanding between key-players, bad practice in regard to input data, inadequate model set-up, insufficient calibration/validation, model use outside of its scope, insufficient knowledge on some processes, miscommunication between the modeller and the end-user, overselling of model capabilities, confusion on how to use model results in decision making and a lack of documentation and transparency of the modelling process[31].

An additional complicating factor is related to the changing character of model-based problem solving projects from monodisciplinary, single person and academic oriented research model studies into multidisciplinary, decision support oriented projects, in which teams consisting of members with different background and different roles have to cooperate to complete the complex job. Modelling in multidisciplinary modelling teams facilitates exploring more complex questions, but this also makes cooperation in such teams more difficult. Team members with different scientific backgrounds encounter more communication problems, which makes managing multidisciplinary model-based water management projects a cumbersome affair (Scholten and Kassahun, 2006, Scholten *et al*., 2006).

The European Commission funded the HarmoniQuA[32] project aimed at lowering many of the hurdles encountered in present simulation oriented modelling by providing modelling guidelines, structured in a knowledge base and by developing a tool to support projects that use models for problem-solving (Scholten *et al*., 2007). The context of the HarmoniQuA project, which involves 12 partners in 10 countries and 10 different languages, introduced new problems. These were partly associated with language issues, which were beyond the scope of the project as professional modellers were assumed to understand English sufficiently and partly because of the variety of modelling cultures in various countries ranging from very professional and mature in northwest Europe to novel and ad hoc in some central and south European countries. These discrepancies in expertise were further enhanced by the level of maturity of the scientific disciplines behind the water domain models, ranging from very mature for groundwater modelling to immature for ecological and socio-economic models (Refsgaard *et al*., 2005).

More details, objectives and major scientific achievements are given in the Appendix B and in Scholten *et al*. (2007), while the aspects relevant for this book are presented and discussed in Chapter 6 and Chapter 7.

[31] Similar problems have been found in many larger projects since 1970's, e.g. modelling efforts for the Great lakes.
[32] HarmoniQuA is an acronym for *Harmonizing Quality Assurance in model-based catchment and river basin management*. From 1 January 2002 to 31 December 2005, this research project was supported by the European Commission under the Fifth Framework Programme and contributed to the implementation of the Key Action "Sustainable Management and Quality of Water" within the Energy, Environment and Sustainable Development Programme. Contract n°: EVK1-CT-2001-00097.

2.5.6 Structured modelling and simulation as building block

In this book I will heavily rely on previous achievements of structured modelling, as described by Scholten (1994) and briefly discussed in section 2.5.2, the GMP Handbook (section 2.5.3) and formal norms (section 2.5.4), but mainly on the results of the HarmoniQuA project (section 2.5.5).

2.6 Relational databases

2.6.1 Outline of the relational database model

In the past decades a data management revolution occurred in response to severe problems with data reusability associated with file-processing approaches to application development. The need to share data resources resulted in the development of database management systems (DBMS), which separate the data from the applications that use the data. Advances in database technology have been boosted primarily by the development, refinement, and eventual implementation of the *relational data model*. DBMSs enable the efficient sharing databases, tools and services for data analysis that are developed and supplied by various providers and distributed on LAN or Internet.

The relational data model for database management is a database model based on predicate logic and set theory. It was first formulated and proposed by Edgar Codd (Codd, 1970) aiming to avoid, without loss of completeness, the need to write computer programs to express database queries and enforce database integrity constraints. Later others have been extended and completed the relational model, e.g. Date (2004).

2.6.2 Relational model as building block

The relational model is here used as an example. Model-based problem solving needs a similar formalization in order to enable a more comprehensive model analysis, using approaches from various paradigms and enabling involvement of modellers (and others) with different paradigmatic backgrounds. Moreover the relational model resembles to some extent the ontological approach chosen in this book and discussed in Chapter 3 and Chapter 4.

2.7 Knowledge-based systems and Artificial Intelligence

2.7.1 Outline of knowledge-based systems and Artificial Intelligence

More than 20 years ago Elzas described how modelling and simulation could benefit from artificial intelligence (AI) and knowledge engineering (KE) (Elzas 1986a,b,c). Especially concepts for structuring and representing knowledge are still innovative for modelling and simulation. These include: classes, rules, frames, semantic nets, model management using a model base, support in model creation, model behavior support, domain knowledge bases with pieces of relevant knowledge and or (sub)models, experimental design aids to support simulation experiments, automatic model generation, input data management systems, a simulation experiment supervisor. Elzas also developed a useful structure for representing knowledge for modelling and simulation, called the *entity structure concept* and associated software (ESP), which will be discussed briefly in Chapter 6 (Elzas, 1989).

In this domain several subdisciplines are involved[33]: artificial intelligence (AI), knowledge based systems (KBS), expert systems (ES) and knowledge engineering (KE). Besides these conventional AI schools of thought, *computational intelligence* provides AI related problem-solving and computing methods, e.g. neural

---

[33] Computer User High-Tech Dictionary (http://www.computeruser.com/resources/dictionary) defines these terms as follows:

> **AI**: *intelligence that mimics human intelligence, when exhibited by devices and applications such as robots or computers with voice recognition and language processing ability. This human-like intelligence implies the ability to learn or adapt through experience.*
> **KBS**: *a computer system that is programmed to imitate human problem solving by means of artificial intelligence and reference to a database of knowledge on a particular subject.*
> **ES**: as *a computer system that is programmed to imitate the problem-solving procedures that a human expert makes.*
> **KE**: *the design and development of knowledge-based systems.*

nets, fuzzy systems and evolutionary computing (incl. genetic algorithms), which are not within the scope of this book.

A comprehensive methodology for knowledge engineering is CommonKADS[34] (Schreiber *et al*., 1999). CommonKADS integrates knowledge management[35], knowledge analysis and knowledge engineering into knowledge systems. Recent developments in AI related fields are focused on the representation, use and reuse of knowledge, e.g. ontologies and ontology formats (OWL[36]), the Semantic Web[37], WordNet's lexical reference system[38] and the Gene Ontology[39] for genomics. Ontologies will be discussed in Chapter 3.

2.7.2 AI as building block

The model-based problem-solving framework proposed in this book does not aim to realize an expert system in the classical sense. The framework focuses on combining the strength of computers with that of human experts in the field of modelling and application domain experts whose knowledge is included in the simulation models. Computers are fast, logical and precise (if fed with the right software and data) and human experts are intelligent and have a more creative way of thinking. Here an ontological approach has been chosen. Ontologies are the products of state-of-the-art knowledge engineering technology to structure knowledge and make it explicit. Ontologies will be discussed in more detail in Chapter 3.

2.8 Process defining technologies

2.8.1 Introduction to process defining technologies

This section 2.8 summarizes of process defining technologies. Defining processes is an essential part of this book, as model-based problem solving is seen as a process.

The following topics will be briefly discussed here: (some) process defining languages, workflow management, good modelling practices and other modelling guidelines and project management. The section ends by choosing the process defining technology that will be used in this book.

2.8.2 Some process defining languages

Zur Muehlen and Becker (1999) evaluate some major process definition languages (PSL, WPDL, PIF, GPSG and UML). Their paper is an excellent introduction to these languages. In the sequel I will give a brief overview of PSL, Petri Nets, PDDL, PIF and UML in order to compare these approaches with the one described further on in this chapter.

The *Process Definition Language* (PSL) aims at creating a process representation that is common to all manufacturing applications, generic enough to be decoupled from any given application and robust enough to be able to represent the necessary process information for any given application. Such a common representation facilitates communication between various applications and communication between different organizations that need to work together for a specific project. Many scientific publications (e.g. Schlenoff *et al*., 1996, Bock and Grüninger, 2004, Grüninger, 2004) discuss aspects of PSL[40]. PSL is being standardized within ISO[41].

---

[34] See: http://www.commonkads.uva.nl/.
[35] Wikipedia (http://en.wikipedia.org/wiki/Knowledge_management): Knowledge Management (KM) refers to a range of practices used by organizations to identify, create, represent, and distribute knowledge for reuse, awareness and learning across the organization.
[36] OWL (Wikipedia): http://en.wikipedia.org/wiki/Web_Ontology_Language.
[37] Semantic Web (Wikipedia): http://en.wikipedia.org/wiki/Semantic_web.
[38] WordNet (Wikipedia): http://en.wikipedia.org/wiki/WordNet or http://wordnet.princeton.edu/.
[39] Gene Ontology (Wikipedia): http://en.wikipedia.org/wiki/Gene_Ontology.
[40]A more complete overview on PSL and publications on PSL can be found from http://www.mel.nist.gov/psl/.
[41]Joint Working Group 8 of Sub-committee 4 (Industrial data) and Sub-committee 5 (Manufacturing integration) of Technical committee ISO TC 184 (Industrial automation systems and integration), see http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/PSL_(18629)/

Petri Nets are a formal and graphically appealing notation method, which is appropriate for modelling systems with concurrency and resource sharing. Petri Nets have been under development since the beginning of the 1960'ies (Petri, 1962). The method is a generalization of automata theory such that the concept of concurrently occurring events can be expressed. Several extensions, e.g. colored Petri nets (Jensen, 1997a, b, c, Kristensen *et al.,* 1998, Van der Aalst, and Van Hee, 2004), have enriched the representation power of Petri nets. Petri nets are used for a multitude of application areas, of which describing processes in general and workflow management applications (see subsection 2.8.3) are relevant in this context.

Planning Domain Definition Language (PDDL) and its successor OPT (Ontology with Polymorphic Types) were developed for deterministic planning domains and problems (Ghallab *et al*., 1998).

WPDL (Workflow Process Definition Language) introduced by the Workflow Management Coalition (WfMC, see also section 2.8.3) is based on a standardization of terms used in the context of workflow management applications[42]. It consists of a reference model and a set of interfaces for (1) process definition services, (2) workflow client applications, (3) invoked applications, (4) other workflow enactment services and (5) administration and monitoring services. WPDL has been mapped on an XML-format, XPDL (XML Process Definition Language)[43]. This language does not define how to install the process definition into the engine. That is the role of Wf-XML, which is a web services protocol that can be used to address a process engine remotely for the purpose of sending or retrieving the process definitions.

The Process Interchange Framework (PIF) was developed as a standardized language for the processes recorded in the MIT Process Handbook project (Lee *et al.*, 1998), which aims at collecting ‘usual’ business process to advice in selecting alternative processes for redesign of existing processes. All constructs in PIF are specified in the Knowledge Interchange Format (KIF)[44]. A process description in PIF is based on a set of frame definitions. Each of these frame definitions denotes an entity type that can be instantiated and arranged in a hierarchy. Values of an attribute within a frame may refer to another frame. In this way relations between instances of the frames can be represented, yielding a resemblance with an ontological representation.

Opposite to the previous process definition languages, which are based on textual process descriptions, the Unified Modelling Language (UML) uses diagram types for the design of object oriented software systems[45]. Each type of diagram gives a different view on a system. The concepts of UML can be extended or specialized by users. The different diagram types together allow users to define of a (workflow) process (Jacobson, *et al*., 1998).

2.8.3 Workflow management

The world of workflow management (at present, often referred to as ‘Business Process Management’, BPM) is completely separated from the world of multidisciplinary model based problem solving, at least in the perception of the professionals and researchers in both worlds. They hardly realize that they share at least the overlapping area of process management in general. The processes of both worlds are to a great extent different, but both try to manage the processes by defining them as formally as possible and with help of computer systems. In both cases, actors (i.e. users) are confronted with similar problems, similar approaches to solve these problems and similar types of solutions. A good state-of-the-art report on workflow management can be found in the Workflow Handbook (e.g. Fischer, 2003) that is annually published, while a more critical review can be found in Dourish (2001). An initiative to improve and support multidisciplinary model based problem solving can benefit from the ideas and designs developed in the world of workflow management and therefore workflow management will discussed here briefly.

Most authors on workflow management distinguish two major parts of a workflow management system: the *process definition component* and a *run-time component* (Fischer, 2003, Prior, 2003). The same two components can also easily be recognized in a modelling support system, but the variety in and dynamics of

[42] I refer here to http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf (WFMC-TC-1011, Feb-1999, 3.0).
[43] Recent documents on WfMC standards and interfaces can be found on http://www.wfmc.org/.
[44] Draft proposed American National Standard (NCITS.T2/98-004): http://logic.stanford.edu/kif/dpans.html.
[45] Up-to-date information can be found from http://www.uml.org/.

process definitions of business processes can obviously be richer than that of modelling, where the process structure is more fixed or at most a variation of some basic pattern. Only if one aims at some generic system to support the design and execution of generic processes, the process definition component should be very flexible to facilitate (re)designing and editing processes. The run-time component requires many features. Adler (2003) classifies these functionalities in four groups. The first group consists of *document management* features, providing storage and retrieval of all kinds of (electronic) documents. The second group is composed of *workflow management* (in the narrow sense of the word) enabling co-operation of groups of participants and controlling the flow of activities in a (business) process. Workflow features and document management support 70 % of the workload, which is definable and therefore structured. The third group consists of *groupware*, including email, databases, shared document databases and electronic forums. Most of these features can at present be found in group co-operation software like ProjectPlace[46], QuickPlace[47], the suite of Lotus Notes and Domino[48] and many others, e.g. the ACE project[49]. The fourth group consists of *knowledge management* functionality to support collecting, validating and structuring knowledge during a (business) process. In general any Business Process Management system should allow distributed (ubiquitous), asynchronous co-operation between groups working in the same process. Business process management consists of activities in three categories: *design*, *execution* and *monitoring*, but the same applies for most processes in which persons have to cooperate in order to complete some large and complex job.

2.8.4 Good Modelling Practices and other modelling guidelines

In section 2.5.3 the (Dutch) Good Modelling Practice Handbook has been presented and discussed briefly. It is intended to promote transparency and repeatability of modelling projects for water management and to facilitate an efficient and effective audit by defining the process. This definition is subsequently used as guideline to help modellers find their way in the maze formed by the complex process of simulation and modelling for water management. The definition is in plain English (originally in plain Dutch), but structured in an informal, although consistent way (Van Waveren *et al*., 1999, Scholten, 1999, Scholten *et al*., 2000, Blind *et al*., 2000, Scholten *et al*., 2001, Scholten, 2001, Scholten and Osinga, 2001, Van der Molen *et al*., 2002).

The GMP Handbook is an informal ontology (Uschold and Grüninger, 1996), as it is loosely expressed in a natural language (see Chapter 3). The GMP approach shares with ontologies that there is wide agreement on its content and, moreover, that it facilitates communication on modelling for water management in general and auditing modelling projects in particular. But it lacks many of the advantages of (more) formal ontologies. The GMP Handbook is a book, i.e. has a sequential structure. Therefore, it is difficult to find all guidance on tasks to perform, including when filling in forms that record what has been done. A second group of shortcomings in the GMP approach is connected with the ease of changing and updating. This refers to changing the structure (i.e. the sequence of the tasks, adding new tasks, deleting tasks) and to changing the description of tasks (what to do, how to do it, methods, pitfalls and sensitivities). This inflexibility did obstruct all plans for regular updates of the GMP Handbook.

The set of more formal norms, discussed in section 2.5.4 is basically the same as the GMP Handbook, but substantially improved in transparency and consistency. It can be seen as a new improved version and has more or less the same guiding functionality for modellers in water management. Furthermore, the norms have an official status, but law does not automatically enforce them. It is up to the client of a modelling project to demand the modelling organization to follow these norms and guidelines.

Next to the Dutch initiative for modelling guidelines in water management, there are several other guidelines for modelling in water management. Refsgaard and Henriksen (2004) review philosophical aspects and terminology and Refsgaard *et al*. (2005) discuss guideline typology, both related to modelling for water management. The GMP Handbook is an example of what they call the ‘public interactive guidelines’ (i.e. *established through a public consultative and consensus building process with an additional focus on regulating the interaction between the modeller and the water manager, who often have the roles of consultant and client, respectively*). Some of the other examples will be discussed here briefly.

---

[46] See www.ProjectPlace.com.
[47] At present called Lotus Team Workplace, see www.lotus.com.
[48] See www.lotus.com.
[49] See www.ace.com.

Modelling guidelines for the Murray–Darling Basin in Australia were developed due to the perception among end-users that model capabilities may have been ‘over-sold’, and that there was a lack of consistency in approaches, communication and understanding among and between the modellers and the water managers, which often resulted in considerable uncertainty for decision making (Middlemis, 2000). As pointed out by Merrick *et al*. (2002) good modelling practice cannot be decomposed into a set of rigid rules that can be followed without communication between modellers and water managers. Furthermore, there is a risk that modellers will not embrace guidelines aiming to inject too much consistency in the review procedure. Experiences from Australia have shown that water managers (non-modellers) interpret review reports commonly as quite negative. This may mostly be the case for projects where there has not been a proper specification of the purpose and conditions at the initiation of the modelling study or where previous reviews during earlier project stages have been inadequate.

The Danish Handbook of Good Modelling Practice and draft guidelines (Henriksen 2002) are similar to the Australian ones, although some important details differ. The water managers, who also ensure that they presently are being used in most studies, have initiated the Danish guidelines. The Danish guidelines include communication between modellers and water managers.

Quality Assurance guidelines are generally very well developed in the UK. Application of guidelines is prescribed as a routine in most areas of model application. Thus, in general the UK market for modelling services is well regulated and characterized as being mature. Most of the guidelines are without clear interactions between modeller and water manager except for some recent ones. The exceptions to this are the surface water quality and biota (ecological) domains where no general guidelines exist. The guidelines in these domains are therefore confined to internal procedures inspired by textbooks and manuals (Packman, 2002).

Californian guidelines prepared by Bay-Delta Modelling Forum (BDMF, 2000) provide a framework, but with very few technical details. The main emphasis of these guidelines is on the interaction between modellers, managers and the public. In this respect various kinds of reviews are prescribed at various stages of the modelling process.

The American guidelines described in ASTM (1992, 1994) are especially comprehensive in the groundwater domain, where they have served as inspiration for all other groundwater guidelines, including the Australian and the Danish guidelines. There are a number of guidelines on various elements of the modelling process. These guides are more than 10 years old and are mainly of technical nature, while limited focus is put on the interaction and review process.

In addition to the above QA guidelines ISO (the International Organization for Standardization) regularly publishes quality management and quality assurance standards. ISO standards provide guidance on the fundamental principles and procedures, but on a rather general level. There are ISO standards addressing development, supply and maintenance of computer software (ISO 9000-3:1997)[50] and other standards providing guidance for a general process based quality management system in an organization (ISO 9004:2000)[51]. However, none of the ISO standards include any particular guidance on matters related to water resources modelling or management, and they are therefore of limited practical use as compared to the above other QA guidelines specialized for water resources modelling.

2.8.5 Project management

Project management is the management of responsibilities, liaisons, quality and people (van der Weide, *et al*. 2003) within the constraints of time, cost and scope. Project management is often related to large, one-off projects, e.g. constructing a building, launching a new product. A scientific research project is usually a one-off process too and its management can therefore be compared with ‘normal’ project management. In opposition to these one-off projects, small, everyday processes in an office environment fall within the scope of workflow management. These processes are (relatively) small but have to be performed many times, are

[50] At present the standards ISO 9001, ISO 9002 and ISO 9003 have been integrated into the new ISO 9001:2000.
[51] http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28692.

often the work of teams, have specific tasks sequences and have time-sensitive deadlines. Typically companies that use workflow management have many hundreds or even thousands of these small processes going on at the same time.

Workflow management applications and the modelling support tool, discussed in Chapter 7, share some aspects with project management software. All three have an initial stage, in which the project/process is defined, i.e. in project management terms called ‘planning’. Dependencies (between parts) and other structural aspects are treated differently in the three approaches, although all have to define them individually. During project/process execution the support provided looks rather different, but is similar to some extent. The project status can be supervised in all approaches and this is one of the many features shared in project management and workflow management. But project management does not facilitate appropriate guidance during project/process execution.

2.8.6 Process defining technologies as building bock

Not many concepts of existing process defining technologies will be used in the remainder of this book. Instead of joining an existing process definition method a new approach will be chosen in developing a framework for model-based problem-solving. This framework will rely on to developments discussed in this section. Its structure will use an ontological knowledge representation format (section xx and Chapter 3), while its modelling guidance content will build on top the Dutch GMP Handbook (and the norms derived from it) and the Australian guidelines for groundwater modelling.

2.9 Software engineering and process/modelling support tools

2.9.1 Introduction to quality assurance for software engineering

Many Quality Assurance approaches for software engineering regard the development of software as a process that has to be improved. A large part of the scientific work necessary for this book has been performed in the context of the HarmoniQuA project (see section 2.5.5). The main activities in this project consisted of developing an ontological guideline for model-based water management and developing a software tool to support the creation of these guidelines and a software tool to support teams in their work for modelling projects. Both tools have been developed in accordance with state-of-the-art software engineering approaches, which will be discussed here briefly.

2.9.2 Design methods of software engineering

The software developments can be divided in two main groups: those following a *waterfall model*[52] and *iterative methods* following an *incremental model*. One of the iterative methods is RAD[53], i.e. Rapid Application Development, which has short, repetitive cycles instead of a single one. It consists of the following core elements: *prototyping*, *iterative development*, *time boxing*, *limited number of experienced and flexible team members*, *a RAD kind of management approach* and it often uses specific *RAD tools*.

Many recent software development approaches are also iterative, resembling RAD to some extent, e.g. *Agile Software Development* and *Extreme Programming*.

2.9.3 Software engineering as building block

The design of the tools for the HarmoniQuA project, proposed in this book, followed a stepwise approach, based on RAD. Many *prototypes* have been used to get feedback of users. In an *iterative development* new features have been added continuously (i.e. 3 major releases and 10 minor releases). This rhythm was enforced by the requirements of the European Commission (EC) for research projects and by practical considerations related to the frequency of meetings of the software development group. *Time boxing* aims at spending a limited amount of time/resources and see what can be achieved within this amount. In this way, one always has some intermediate product, but a disadvantage is the implicit mechanism to reduce the

---

[52] The waterfall model can be defined as: a sequential software development model (a process for the creation of software) in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance (Wikipedia: http://en.wikipedia.org/wiki/Waterfall_model).
[53] RAD was originally introduced by James Martin.

number of features to meet time box requirements. The next aspect of RAD, i.e. a *limited number of experienced and flexible team members*, has been realized in the HarmoniQuA project, where a few developers were responsible for the development and a few other persons, not involved in the development, for testing of the software. The software development followed a RAD kind of *management approach* with short development cycles, enforced deadlines, a motivated team and focus on lowering bureaucratic hurdles. RAD tools have been used in the development of MoST[54] and its associated tools, including a Java development environment, cooperation software and a version control system, all intended to facilitate and speed up software development.

2.10 Building blocks as a fundament for a model-based problem solving framework

The building blocks discussed in this chapter are assumed as pre-existing knowledge, on top of which new framework for model-based problem solving can be built. The building blocks originate from various paradigms and provide a vocabulary from these paradigms, as well as theories, methods and practices. The building blocks act as a fundament of a framework for model-based problem solving that will be constructed further using ontologies and tools to define and support problems, models and modelling.

The new framework has to result in better modelling practices and improve the potential of model-based problem solving by facilitating the exchange of paradigmatic objects, e.g. concepts, methods, solvers and practices.

2.11 References

Ackoff, R.L., 1960. Systems, organizations, and interdisciplinary research, General Systems Yearbook, vol. 5, Society for General Systems Research, pp. 1-8.

Adler, M., 2003. Group applications: from workflow to work management. In: Fischer, L. (Ed.), Workflow Management 2003. Future Strategies Inc., Book Division, Lighthouse Point, FL, USA, pp. 27-38.

ASTM, 1992. Standard practice for evaluating mathematical models for the environmental fate of chemicals, American Society for Testing and Materials, Report Standard E978-92.

ASTM, 1994. Standard guide for application of a ground-water flow model to a site-specific problem, American Society for Testing and Materials, Report Standard D5447-93.

BDMF, 2000. Protocols for Water and Environmental Modelling, Bay-Delta Modelling Forum. Ad hoc Modelling Protocols Committee. www.sfei.org/modellingforum/, Report.

Beck, M.B., 1987. Water quality modelling: a review of the analysis of uncertainty. Water Resources res. 23, 1393-1442.

Beer Stafford, A., 1972. Brain of the Firm: The Managerial Cybernetics of Organization. Allen Lane, Penguin Press, London, First Edition, ISBN 0471276871.

Beulens, A.J.M. and H. Scholten, 2000. A view on modelling paradigms: a prerequisite for Good Modelling Practice. Extended Abstracts of 14th JISR-IIASA Workshop on Methodologies and Tools for Complex Systems Modelling and Integrated Policy Assessment, July 3-5 2000, IIASA, Laxenburg, Austria.

Beulens, A.J.M., and H. Scholten. 2001. Model solving software and modelling paradigms: a challenge for developers. Pages 4-8 in M. Makowski, editor. Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modelling and integrated policy assessment, August 27-29, 2001, IIASA, Laxenburg, Austria.

Blind, M.W., H. Scholten and R.H. van Waveren, 2000. Model for success. Water21, October 2000, 43-45.

Bock, C., Grüninger, M., 2004. PSL:A semantic domain for flow models. Software & Systems Modelling 4, 209-231.

Boulding, K.E., 1956. General Systems Theory: the skeleton of science. Management Science 2, 197-208.Cannon, W.B., 1932. The wisdom of the body. W. W. Norton, New York.

Checkland, P.B. and J. Scholes, 1990. Soft systems methodology in action. John Wiley & So Ltd.ns, Chichester, 329 pp.

Checkland, P.B., 1981. Systems thinking, systems practice. John Wiley & So Ltd.ns, Chichester, 330 pp.

Codd, E.F., 1970. A relational model of data for large shared data banks. Communications of the ACM 13, 377-387. http://www.acm.org/classics/nov95/toc.html.

Crosby, P., 1979. Quality is free: the art of making quality certain. McGraw-Hill Book Company, New York.

[54] MoST is a Modelling Support Tool discussed in Chapter 7.

Davis, L. and S. Coombs, 1989. Optimizing network link sizes with genetic algorithms. In: M.S. Elzas, T.I. Oren, B.P. Zeigler (Eds.), Modelling and simulation methodology, Modelling and simulation, vol., North-Holland, Amsterdam, pp. 317-331.
Date, C.J., 2004. An introduction to database systems. Addison-Wesley Systems Programming Series. Addison-Wesley Publishing Company, Reading etc., 8th Edition, ISBN ISBN 0321197844.
Davis, L., 1987. Genetic algorithms and simulated annealing. Pitman, Londoned., 216 pp.
de Wit, C.T., 1960. On competition. Verslag Landbouwkundig Onderzoek, nr. 66.8. PUDOC, Wageningen.
de Wit, C.T., 1965. Photosynthesis of leaf canopies. Verslag Landbouwkundig Onderzoek, nr. 663. PUDOC, Wageningen.
Dourish, P. 2001. Process descriptions as organisational accounting devices: the dual use of workflow technologies. Pages 52-60 in Proceedings of the 2001 International ACM SIGROUP Conference on Supporting Group Work, October 2001. ACM, Boulder, Colorado.
Elzas, M.S., 1980, Simulation and the Processes of Change. In: T.I. Ören, C.M. Shub, P.F. Roth (Eds.), IEEE Conference, Simulation with discrete models: a state-of-the-art view, Orlando, FL, USA, IEEE, New York, 3-18.
Elzas, M.S., 1984. System paradigms as reality mappings. In: T.I. Ören, B.P. Zeigler, M.S. Elzas (Eds.), Simulation and model-based methodologies: an integrative view, NATO ASI Series F: Computer and system science, vol., Springer-Verlag, Berlin, ISBN, pp. 41-67.
Elzas, M.S., 1986a. The kinship between artificial intelligence, modelling & simulation: an apraisal. In: M.S. Elzas, T.I. Ören, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era, vol., Elsevier, Amsterdam, ISBN 0-444-701303, pp. 3-13.
Elzas, M.S., 1986b. The applicability of artificial intelligence techniques to knowledge representation in modelling and simulation. In: M.S. Elzas, T.I. Ören, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era, vol., Elsevier, Amsterdam, ISBN, pp. 19-40.
Elzas, M.S., 1986c. Relation between artificial intelligence environments and modelling &. In: M.S. Elzas, T.I. Ören, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era, vol., Elsevier, Amsterdam, ISBN 0-444-701303, pp. 61-77.
Elzas, M.S., 1989. The role of artificial intelligence/entity structure concept manipulation methods in system modelling and design. In: M. Elzas, T. Ören, B. Zeigler (Eds.), Modelling and simulation methodology. Elsevier Science Publishers B.V., Amsterdam, New York, Oxford, Tokyo, pp. 3-27.
Elzas, M.S., T.I. Ören and B.P. Zeigler, 1989. Modelling and Simulation Methodology: Knowledge Systems Paradigms USA, Elsevier Science Inc., New York.
Fischer, L., 2003. Workflow Handbook 2003. Future Strategies Inc., Book Division, Lighthouse Point, FL, USA, 362 pp.
Forrester, J.W., 1961. Industrial dynamics. Pegasus Communications., Waltham, MA, 464 pp.
Gardner, R.H. and R.V. O'Neill, 1983. Parameter uncertainty and model predictions: a review of Monte Carlo results. In: M.B. Beck, G. Van Straten (Eds.), Uncertainty and forecasting of water quality, vol., Springer Verlag, New York, pp. 245-257.
Gardner, R.H., B. Rojder and U. Bergstrom, 1983. PRISM: A systematic method for determining the effect of parameter uncertainties on model predictions. Studsvik.
Gardner, R.H., R.V. O'Neill, J.B. Mankin and J.H. Carney, 1981. A comparison of sensitivity analysis and error analysis based on a stream ecosystem model. Ecol. Modelling 12, 173-190.
Gardner, R.H., W.G. Cale and R.V. O'Neill, 1982. Robust analysis of aggregation error. Ecology 63, 1771-1779.
Geoffrion, A.M., 1987. An introduction to structured modelling. Management Science 33, 547-588.
Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., Wilkins, D., 1998. PDDL—the planning domain definition language, Yale Center for Computational Vision and Control, Report Technical Report CVC TR-98-003/DCS TR-1165.
Goldberg, D.E., 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company Inc., Reading, Massachusetts etc.ed., 412 pp.
Goldberg, D.E., 1994. Genetic and evolutionary algorithms come of age. Communications of the ACM 37, 113-119.
Grüninger, M., 2004. Ontology of the Process Specification Language. In: S. Staab, R. Studer (Eds.), Handbook of ontologies. Springer, pp. 575-592.
Henriksen, H., 2002. Danish groundwater modelling guidelines. In: J. Refsgaard (Ed.), State-of-the-Art Report on Quality Assurance in modelling related to river basin management. Geological Survey of

Denmark and Greenland, Copenhagen. Chapter 14, http://harmoniqua.org/public/Reports/SOA%20chapters/SOA.pdf.
Humphrey, W.S., 1989. Managing the software process. Addison-Wesley, Reading etc., 494 pp.
Jacobson, I., G. Booch and J. Rumbaugh, 1998. The Unified Software Development Process, Addison Wesley Longman, Boston, ISBN 0-201-57169-2., 423 pp.
Jensen, K., 1997a. Coloured Petri Nets. Volume 1, Basic Concepts. Springer-Verlag, London, 2nd corrected printing, ISBN: 3-540-60943-1, 234 pp.
Jensen, K., 1997b. Coloured Petri Nets. Volume 2, Analysis Methods. Springer-Verlag, London, 2nd corrected printing, ISBN: 3-540-58276-2, 174 pp.
Jensen, K., 1997c. Coloured Petri Nets. Volume 3, Practical use. Springer-Verlag, London, ISBN: 3-540-62867-3, 265 pp.
Kassahun, A., H. Scholten, G. Zompanakis and C. Gavardinas, 2004. Support for model based water management with the HarmoniQuA toolbox. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Proceedings of iEMSs 2004: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004, pp. 1282-1287.
Klir, G.J., 1985. The Architecture of Systems Problem-solving. Plenum Press, New York.
Knepell, P.L. and D.C. Arangno, 1993. Simulation validation: a confidence assessment methodology. IEEE Computer Society Press, Los Alamitos, CA, 150 pp.
Köhler, W., 1924. Die physischen Gestalten in Ruhe und im stationären Zustand: Eine naturphilosophische Untersuchung. Verlag der Philosophichen Akademie, Erlangen.
Köhler, W., 1927. Zum Problem der Regulation. In: M. Henle (Ed.), The selected papers of Wolfgang Köhler (1971), Liveright, New York, pp. 305-326.
Kristensen, L., Christensen, S., Jensen, K., 1998. The Practitioner's Guide to Coloured Petri Nets. International Journal on Software Tools for Technology Transfer 2, 98-132.
Lee, J., M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Yost and other members of the PIF Working Group, 1998. The PIF Process Interchange Format and Framework Version 1.2. The Knowledge Engineering Review 13, 91-120.
Lotka, A.J., 1920. Undamped oscillations derived from the law of mass action. J. Am. Chem. Soc. 42, IS95-I599.
Lotka, A.J., 1925. Elements of physical biology. Williams and Wilkins, Baltimore.
Markestijn, J., 2004. Communities of practice and supporting software, Artificial Intelligence, Knowledge Management and Knowledge Technology, Faculty of Sciences. Vrije Universiteit, Amsterdam, 97 (+XV).
Maturana, H.R. and F.J. Varela, 1973. De máquinas y seres vivos: Una teoriá de la organizacíon biológica. Editorial Univeristaria, Santiago (Chili).
McCall, J.A., P.K. Richards and G.F. Walters, 1977. Factors in software quality. National Techical Information Service, Springfield, Va.
Meadows, D., W.W. Behren III, D. Meadows, R.F. Naill, J. Randers and E.K.O. Zahn., 1974. Dynamics of Growth in a Finite World. Wright-Allen Press, Cambridge.
Meadows, D.H., D.L. Meadows, J. Randers and W.W. Behrens III, 1972. The Limits to Growth: A Report for the Club of Rome's Project on the Predicament of Mankind. Universe Books, New York, ISBN 0-87663-165-0.
Merrick, N., H. Middlemis and J. Ross, 2002. Groundwater modelling guidelines for Australia - the review process., International Association of Hydrogeologists Groundwater Conference “Balancing the Groundwater Budget”, May, 2002., Darwin, Australia.
Middlemis, H., 2000. Murray-Darling Basin Commission. Groundwater flow modelling guideline., Aquaterra Consulting Pty Ltd., Report Project no. 125.
Miller, J.G., 1978. Living Systems. Mcgraw-Hill (Tx), New York, ISBN ISBN: 0070420157, 1102 pp.
Minsky, M.L., 1965. Matter, mind, and models, Proceedings of International Federation of Information Processing Congress, vol. 1, Spartan Books, pp. 45-49.
O'Neill, R.V. and R.H. Gardner, 1979. Sources of uncertainty in ecological models. In: B.P. Zeigler, M.S. Elzas, G.J. Klir, T.I. Oren (Eds.), Methodology in systems modelling and simulation, vol., North-Holland Publ. Co., Amsterdam, pp. 447-463.
O'Neill, R.V., R.H. Gardner and J.B. Mankin, 1980. Analysis of parameter error in a nonlinear model. Ecol. Modelling 8, 297-311.
O'Neill, R.V., R.H. Gardner and J.H. Carney, 1982b. Parameter constraints in a stream ecosystem model: incorporation of a priori information in Monte Carlo error analysis. Ecol. Modelling 16, 51-65.

O'Neill, R.V., R.H. Gardner, L.W. Barnthouse, G.W. Suter, S.G. Hildebrand and C.W. Gehrs, 1982a. Ecosystem risk analysis: a new methodology. Environmental Toxicology and Chemistry 1, 167-177.
Oreskes, N., K. Shrader-Frenchette and K. Belitz, 1994. Verification, Validation,and Confirmation of Numerical Models in the Earth Sciences. Science 263, 641-646.
Packman, J.C., 2002) Quality Assurance in the UK. In: Refsgaard, J.C. (Ed) State-of-the-Art Report on Quality Assurance in modelling related to river basin management. Chapter 17, Geological Survey of Denmark and Greenland, Copenhagen. http://harmoniqua.org/public/Reports/SOA%20chapters/SOA.pdf.
Petri, C., 1962. Kommunikation mit Automaten, Institut für Instrumentelle Mathematik. University of Bonn, Bonn, Schriften des IIM Nr. 2.
Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, 1992. Numerical recipes: the art of scientific computing. Cambridge University Press, Cambridge, 818 pp.
Prior, C., 2003. Workflow and process management. In: Fischer, L. (Ed.), Workflow Management 2003. Future Strategies Inc., Book Division, Lighthouse Point, FL, USA, pp. 17-25.
Refsgaard, J.C. and H.J. Henriksen, 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.
Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.
Rosen, R., 1985. Anticipatory Systems: Philosophical, Mathematical and Methodological Foundations. Pergamon Press, ISBN ISBN 0-08-031158-X.
Rozenblit, J.W. and B.P. Zeigler, 1986. Entity-based structures for model and experimental frame construction. In: M.S. Elzas, T.I. Oren, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era. Elsevier Science Publishers BV (North-Holland), Amsterdam, pp. 79-100.
Rykiel, E., R., 1996. Testing ecological models: the meaning of validation. Ecol. Modelling 90, 229-244.
Sargent, R.G., 1982. Verification and validation of simulation models. In: F.E. Cellier (Ed.), Progress in modelling and simulation, vol., Academic Press, London, etc, pp. 159-169.
Sargent, R.G., 1984a. Simulation model validation. In: T.I. Oren, B.P. Zeigler, M.S. Elzas (Eds.), Simulation and model-based methodologies: an integrative view, NATO ASI Series F: Computer and Systems Sciences, vol., Springer Verlag, Berlin, etc., pp. 537-555.
Sargent, R.G., 1984b. A tutorial on verification and validation of simulation models. In: S. Sheppard, U. Pooch, D. Pegden (Eds.), Proceedings of the 1984 winter simulation conference, vol., pp. 115-121.
Schlenoff, C., Knutilla, A., Ray, S., 1996. Unified Process Specification Language: Requirements for Modelling Process, National Institute of Standards and Technology, Report NISTIR 5910, 80 pp.
Schlesinger, S., R.E. Crosbie, R.E. Gagne, G.S. Innis, C.S. Lalwani, J. Loch, R.J. Sylvester, R.D. Wright, N. Kheir and D. Bartos, 1979. Terminology for model credibility. Simulation 32, 103-104.
Scholten, H., 1994, Quality assurance of the simulation modelling process, A.R. Kaylan, A. Lehmann, T.I. Ören, Proc. of the European Simulation Symposium, Istanbul, October 9-12, 1994, Istanbul, Vol. I, 85-89.
Scholten, H. and A.J. Udink ten Cate, 1995. Testing simulation modelling quality, Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium, pp. II.A.3.1-6.
Scholten, H. and A.J. Udink ten Cate, 1996. Quality assessment of simulation modelling. In: C. Lokhorst, A.J. Udink ten Cate, A.A. Dijkhuizen (Eds.), Information and Communication Technology applications in Agriculture: state of the art and future perspectives. ICCTA, Wageningen, pp. 353-358.
Scholten, H. and A.J. Udink ten Cate, 1999. Quality assessment of the simulation modelling process. Computers and Electronics in Agriculture 22, 199-208.
Scholten, H., 1999. Good Modelling Practice. In: M. Makowski (Ed.), Abstracts of the 13th JISR-IIASA Workshop on methodologies and tools for complex system modelling and integrated policy assessment. IIASA, Laxenburg, pp. 57-59.
Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.
Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, H. Wösten, R.D. Koeze and J.J. Noort, 2000. Good Modelling Practice in water management, Proceedings HydroInformatics2000 (on CD-ROM), vol., International Association for Hydraulic Research, Cedar Rapids, Iowa, USA.
Scholten, H., 2001. An ontological approach in quality assessment of modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001,

SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering. TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0, Delft, The Netherlands.

Scholten, H. and S.A. Osinga, 2001. How to support GMP in model-based DSS. In: M. Makowski (Ed.), Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modelling and integrated policy assessment, August 27-29, 2001. IIASA, Laxenburg, Austria, pp. 45-47.

Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, J.H.M. Wösten, R.D. Koeze and J.J. Noort, 2001. Improving the quality of model-based decision support: Good Modelling Practice in water management. In: A. Schumann, J. Xia, M. Marino, D. Rosbjerg (Eds.), Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the International Association of Hydrological Sciences, Maastricht, The Netherlands, July 18-27, 2001, IAHS Red Books, vol. 268, IAHS, Maastricht, The Netherlands, ISBN, pp. 223-230.

Scholten, H. and S.A. Osinga, 2002. Requirement analysis report of the HarmoniQuA knowledge base and tools. Wageningen University, Wageningen, 102 pp.

Scholten, H., J.C. Refsgaard and A. Kassahun, 2004. Structuring multidisciplinary knowledge for model based water management: the HarmoniQuA approach. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Proceedings of iEMSs 2004: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004, pp. 1288-1293.

Scholten, H. and A. Kassahun, 2006, Supporting multidisciplinary model-based water management projects: a user perspective, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006.

Scholten, H., A. Kassahun and J.C. Refsgaard, 2006. Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006, Research Publishing, Volume 3, 2231-2238.

Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.

Schreiber, G., H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga, 1999. Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, Cambridge (MA), USA, ISBN 0-262-19300-0, 471 pp.

Sheng, G., M.S. Elzas, T.I. Ören and B.T. Cronhjort, 1993. Model validation: a systemic and systematic approach. Reliability Engineering and System Safety Journal 42, 247-259.

Simon, H.A., 1957. Models of Man: Social and Rational. John Wiley and Sons, Inc. New Yorked., 279 pp

Simon, H.A., 1962. The Architecture of Complexity: Hierarchic Systems. Proceedings of the American Philosophical Society 106, 467-482.

Simon, H.A., 1996. The science of the Artificial. MIT Press, Cambridge, London, 3rd ed., 231 pp.

Turban, E., 1995. Decision support and expert systems: management support systems. Prentice Hall, Englewood Cliffs, N.J., 4th, 1005 pp.

Uschold, M. and M. Grüninger, 1996. Ontologies: principles, methods and applications. The Knowledge Engineering Review 11, 93-136.

Van der Aalst, W.M.P. and K.M. Van Hee, 2004. Workflow Management: Models, Methods, and Systems, MIT press, Cambridge, MA, 2nd edition, ISBN 0-262-72046-9.

Van der Molen, D.T., F. Kortstee, J. Noort and H. Scholten, 2002. NEN-norm voor Modelleren in het waterbeheer. H2O 35, 21-22.

Van Straten, G., 1986. Identification, uncertainty assessment and prediction in lake eutrophication, Ph. D. Thesis. Technical University Twente, Febodruk Enschede, 240 pp.

Van Straten, G., 1998. Models for water quality management: the problem of structural change. Wat. Sci. Techn. 37, 103-111.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J.J. Noort, 1999. Good Modelling Practice Handbook. STOWA/RWS-RIZA, Utrecht/Lelystad, original in Dutch (Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice), ISBN 90-5773-056-1, 149 pp.

van der Weide, A., A.J.M. Beulens and S. Van Dijk, 2003. Project Planning and Management., Lemma Publishers, Utrecht, 212 pp.

Verhulst, P.F., 1838. Notice sur la loi que la population pursuit dans son accroissement. Correspondance mathématique et physique 10, 113-121.
Volterra, V., 1926. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. Mem. R. Acad. Nazionale Lincei 2, 31-113.
Von Bertalanffy, K.L., 1934. Untersuchungen über die Gesetzlichkeit des Wachstums. Arch. Entwicklungsmech. 131, 613-652.
Von Bertalanffy, K.L., 1968. General System Theory: Foundations, Development, Applications. George Braziller, New York.
Wiener, N., 1948. Cybernetics or Control and Communication in the Animal and the Machine. Hermann et Cie - MIT Press, Cambridge, MA, Paris.
Wierzbicki, A., M. Makowski and J. Wessels, 2000. Model-based decision support methodology with environmental applications. Kluwer, Dordrecht, Boston, London, ISBN 0-7923-6327-2.
Wymore, A.W., 1967. A Mathematical Theory of Systems Engineering: The Elements. John Wiley & Sons, Inc., New York.
Wymore, A.W., 1976. Systems engineering methodology for interdisciplinary teams. John Wiley & Sons, New York, ISBN 047196901X, 444 pp.
Zeigler, B.P., D. Belogus and A. Boshoi, 1982. ESP- An Interactive Tool for System Structuring, Proc. of the 1980 European Meeting on Cybernetics and Research, Hemisphere Press.
zur Muehlen, Becker, J., 1999. WPDL – State-of-the-Art and Directions of a Meta-Language for Workflow Processes. In: L. Bading *et al*. (Ed.), Proceedings of the 1st KnowTech Forum, September 17th-19th 1999, Potsdam.

3 An ontological approach for dynamic knowledge management

"Science is organized knowledge. Wisdom is organized life."
(Immanuel Kant, 1724 - 1804)

Content of Chapter 3

3.1 Introduction to this chapter

Chapter 1 gave an overview of modelling problems and approaches to solve these, including the development of a framework to support multidisciplinary model-based problem solving. *Modelling* is often a complex process[1], in which one or more persons (teams) have to co-operate in order to represent parts of some *object system*[2] (OS), relevant for the problem at hand in a *model*, and use this model to solve the problem. This framework aims at structuring knowledge on the process of defining and supporting *modelling*, the specification of *problems* (with its *objects systems*) and the definition of one or more appropriate *models*. The complexity of modelling leads to many pitfalls, which modelling teams have to avoid. Furthermore, the perception of modelling is changing in time: science provides new ideas and new methods. Modelling is a part of the scientific methodology in many disciplines and has been so for many years, but each discipline has added its own aspects to the modelling process, making multidisciplinary modelling hard to achieve and hard to manage. Every person involved in modelling knows roughly what has to be done, but this knowledge is partly explicit and partly tacit. This knowledge differs substantially between disciplines, but also within disciplines. The explicit part of the knowledge is not shared by the larger part of the professional modelling community. It is made up of individual, company-wide or group-wide shared practices and routines, which are usually to a large part undocumented or entrusted to the archives of gray literature. Any attempt to make modelling knowledge shared[3] and explicit should be expressed in clear requirements as to the role of this knowledge in the modelling process.

Therefore, the following functional requirements have to be set to a framework to support multidisciplinary model-based problem solving including knowledge on the process of defining and supporting *modelling*, the specification of *problems* (with its *objects systems*) and the definition of one or more appropriate *models*:

- Knowledge should be shared by modelling teams; the teams should have a common understanding and common terminology, at least at interfaces between disciplines;
- Knowledge should be understandable by man and manageable by machines (computers);
- Knowledge is not static but changes in time; this includes modelling methods and tools; furthermore, modelling knowledge should be easy to maintain and easy to improve.

From these functional requirements the following design decision can be taken. The knowledge should be structured in a Knowledge Base (KB), which facilitates maintenance and improvements and which is understandable for man and can be managed by machines. A state-of-the art vehicle for KB design is the concept of ontologies, presently most often used in knowledge engineering[4].

The remainder of this chapter will briefly discuss ontologies and how ontologies can be instrumental in the development of knowledge about the process of defining and supporting *modelling*, the specification of *problems* (with its *objects systems*) and the definition of one or more appropriate *models*. Subsequently, the following sections deal with the *why* and *what* of ontologies, applying ontologies, ontological formats and tools and finally the role of ontologies in multidisciplinary model-based problem solving.

3.2 Why an ontological approach

Twenty years ago the term *ontology* was an esoteric part of philosophy, about being, about what can be mentioned (Gruninger & Lee, 2002). The term originates from Aristotle, who distinguished theoretical philosophy, consisting of *physics*, *ontology* and *logic* and, subsequently practical philosophy, consisting of *ethics*, *politics* and *poetry*. Ontology studies *what is,* as such, its nature, its characteristics and its mutual relations.

About 15 years ago, knowledge engineers introduced a new meaning to the term *ontology*. The most used definition is of Gruber (1993, 1995): *an ontology is an explicit specification of a conceptualization*. Gruber

---

[1] This described in the first draft of a *mental process model* is presented in Chapter 2, Table 2.1.
[2] In Chapter 5 *object system* will be defined as *the things and processes in the real world or some defined system that will be the object of modelling*.
[3] Shared means here: always: shared by the best part of the professional modelling community, which includes model developers and model users.
[4] In fact the term *ontology* has been used in systems science before (Klir, 1985).

(1995) states further, that *a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose*. In an AI-context a *conceptualization* refers to what can be represented in terms of concepts and the relationships among them, reflected in the *representational vocabulary* (Gruber, 1993). Borst (1997) added to this definition that there should be a consensus about what the meaning is of the concepts and the relations between them, resulting in the following definition*: an ontology is a formal specification of a shared conceptualization*. I extend the definition of Borst here to: *an ontology is a concise and precise, formal specification, shared by a group of persons and providing sufficient vocabulary such that a piece of knowledge can be formalized for its purpose, is understandable for its human users and manageable for its machine users (computers)*.[5]

Since the early days of ontologies, about 15 years ago, ontologies have left the AI labs rapidly and moved to the desktops of domain experts. Many of the ontologies found on the Internet are simple, but large taxonomies to categorize websites (e.g. Yahoo). The WWW-consortium, W3C, aims to design an ontological description language to describe knowledge on websites in order to enable web agents to *understand* this knowledge. There are many other applications of ontologies including military, genetic and business applications (e.g. Uschold *et al.*, 1998), terminology applications for products and services, for clinical practice guidelines (e.g. de Clercq *et al*, 2001), for planning in food supply chains (e.g. Houba *et al*., 2000).

As mentioned before, Gruber (1995) has linked ontologies with their *purpose*, which should not be confused with their field of application. Why ontologies are developed is widely described. Here I will follow the scheme of intended purposes for ontologies of Uschold *et al.* (1998), who distinguishes three groups of uses of ontologies: *communication*, *interoperability* and *development of systems*. These groups will be discussed here briefly.

One of the basic aspects of ontologies is that they represent knowledge to be shared and therefore ontologies can be used for *communication* of structured knowledge between people and between organizations (Uschold and Grüninger, 1996). Such a knowledge structure can include, but is not restricted to reference models, large-scale integrated software systems, enterprise structures, complex processes, unambiguous definitions of concepts and terminology, standardization of terminology and the development of an integrative model to facilitate business process reengineering. A second group of uses can be characterized with the term *interoperability*, which means that ontologies enable machine-machine understanding, as is the case in web agents on the Internet. Ontologies can also act as inter-lingua between software components (i.e. to facilitate defining interfaces between the components) and also as repositories for databases (Uschold and Grüninger, 1996). The third group of uses of ontologies refers to the *development of systems*, both software systems as well as knowledge based systems. Ontologies in this case play a role in the specification of such systems and improve the reliability of the systems by facilitating checking the match between requirements and design. The re-use of (components) of such systems and applying these for different domains is also facilitated by ontologies (Uschold and Grüninger, 1996). Further, ontologies help to make assumptions in these systems explicit, which simplifies re-use of the systems.

Some authors (Chandrasekaran *et al.*, 1999) use the word ontology in two senses: for the structure of the knowledge (concepts and relations) and for the knowledge itself. I will use the word in the following way. An *ontology* consists of an *ontological structure* and of an *instance of an ontology*. The instance contains the knowledge itself (i.e. content), while the ontological structure defines how this content is ordered. This two-layered framework (structure and instances) is in practice too simple and one often needs more layers in some hierarchy of ontological layers, the parent providing structure and/or vocabulary to the child in the hierarchy. The relational database paradigm has a similar layered structure with vocabulary in the top layer, data model in the middle layer and the actual data in the lowest layer (Date, 1977).

Making ontologies has normally no purpose on itself, but it is a part of a process to build a knowledge base for some purpose. This process is typically composed of the following steps:

1. An *ontological structure* is made which is the frame of the intended knowledge base;

[5] Compared to the other definitions, two new terms are introduced in this definition: *users* and *purpose*, but these terms are also used by the authors of other definitions (e.g. Gruber, 1993, 1995, Chandrasekaran, 1999) in a similar way, but they did not include them as such in their definitions.

2. A tool based on this ontology is used for *knowledge acquisition*;
3. The acquired knowledge is stored as *instance of the ontology* in a knowledge base;
4. *Software applications* are developed that use this knowledge base.

This approach will be discussed in a next Chapter using the development of a knowledge base and tools to work with it. Here I will summarize what ontologies are and how to apply them.

3.3 What are ontologies?

A comprehensive and clear introduction on what ontologies are and why we need them is given in Chandrasekaran *et al.* (1999). Some terms are essential for ontologies, but how these terms have to be interpreted depends usually on the software used to build the ontology or to work with it. This section will start with the more common idiom and only in the context of specific ontology tools I will use a terminology derived from object-oriented software development, but the terms have different meanings in the world of ontologies.

An ontology can be seen as a framework to represent the structure and content of a body of knowledge on a domain in a formal, machine processable, way. In order to describe the structure and the content of knowledge, ontologies must have one or more standard vocabularies[6] at hand, which define the terms (*concepts*) and *relations* that are used to describe this specific knowledge domain (subject area). To describe a certain piece of knowledge an ontology contains terminology describing the *concepts* and *relations* between them. *Concepts* can be discussed and have to be represented. The term *concept* has thus a broader meaning than *entity* and it encompasses abstract and concrete things, but also processes, tasks and ambitions or goals. *Concepts* are used to define and explain terms. *Relations* organize *concepts* in a hierarchical[7] or in some self-defined structure. Often ontologies contain other elements e.g. *properties*, *functions*, *axioms*, but these are not essential to understand what ontologies are. *Instances*[8] are also parts of an ontology, as they contain actual knowledge. If *task* is a concept in an ontology, the instances of *task* can be *go shopping*, *cook meal*, *eat meal*.

3.4 Applying ontologies

Ontologies can be used for a real hodgepodge of purposes. A first use of ontologies is shown in the next Chapter (Chapter 4) to order the paradigmatic aspects in Kuhn's 'normal' science and in multidisciplinary science. In Chapter 5 an ontological approach will be used to describe a framework to support multidisciplinary model-based problem solving. In the next Chapters ontologies will be discussed in connection with the process of defining and supporting *modelling* (Chapter 6), the specification of *problems* with its *objects systems* (Chapter 8) and the definition of one or more appropriate *models* (Chapter 9).

Some authors distinguish between two types of knowledge suited to be represented in ontologies (Chandrasekaran *et al.*, 1999): *domain factual knowledge* and *problem solving knowledge*. The latter category (problem solving knowledge) is in fact an example of the broader class of process-oriented knowledge common in many application fields, including medical protocols, workflow management, business process redesign, manufacturing industry, software engineering and project management. In the following Chapters this quality of ontologies, suited for structuring many types of knowledge, will be used and knowledge of both types (domain factual knowledge and problem solving knowledge) will be organized in ontologies.

Ontologies can be classified by their degree of formality. An example is the (Dutch) Good Modelling Practice Handbook (Scholten *et al.*, 2000, Scholten *et al.,* 2001), which aims to improve the overall quality of modelling in water management by decomposing a modelling process into steps, as a structured definition

6 The *standard terminology* to describe ontologies is discussed in Chapter 4 (why) and Chapter 5 (what).
7 Wikipedia (http://en.wikipedia.org/wiki/Hierarchy): a hierarchy is a system of ranking and organizing things or people, where each element of the system (except for the top element) is subordinate to a single other element. In taxonomy, a subdiscipline of biology, the *Linnaean taxonomy* is a is a method of classifying living things originally devised by, and named for, *Carl Linnaeus* although it has changed considerably since his time.
8 Chapter 5 defines *instance* as *specialization of a concept*. *Instances* are more particular than *concepts*. *Instances* contain a piece of the knowledge. *A computer*, *a woman* and *a book* are instances of the concepts *computer*, *woman* and *book*.

of the whole process and by providing a glossary of terms used in this process[9]. The GMP Handbook is an example of a completely informal ontology. Uschold and Gruninger (1996) distinguish four degrees of formality, which (explicit) ontologies can have:

1. Highly informal: expressed loosely in natural language, e.g. the GMP Handbook (Van Waveren *et al.*; 1999, Scholten *et al.*, 2000, 2001).
2. Semi-informal: expressed in a restricted and structural form of natural language, greatly increasing clarity by reducing ambiguity, e.g. the text version of the Enterprise Ontology (Uschold *et al.*, 1998)
3. Semi-formal: expressed in an artificial formally defined language, e.g.
   - The Ontolingua version of the Enterprise Ontology (see www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html);
   - The HarmoniQuA guidelines for model-based problem solving in water management implemented in Protégé and available on a web server (see www.HarmoniQuA.org/deliverables.html);
4. Rigorously formal: meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness, e.g. TOVE (Fox, 1992), the PhysSys Ontology (Borst, 1997, Borst *et al.*, 1997).

Although these degrees of formality enrich the vocabulary to discuss ontologies, in reality the categories of Uschold and Grüninger (1996) do not reflect the continuous character of formality actually found in existing ontologies and their instances. The type of user sets requirements to the degree of formality of ontologies. As far as computer programs and web agents have to use (parts of) ontologies, an ontology should be formal, but if human users have to work with, ontologies (at least at the instance level) may be less formal and more directly understandable. Strict formality is always preferable however, as it facilitates re-use of the knowledge and - with the right software - machine 'understanding'[10].

3.5 Ontological formats and editing tools

3.5.1 Developing ontologies

Building ontologies or working with ontologies requires a *format*[11], in which the ontology will be stored and requires preferably a *tool* that helps developing the ontology. There are many formats, some of which will be discussed here briefly.

In the world of knowledge engineers and others there is no agreement on the best way to develop ontologies. Not even on what are exactly ontologies. Knowledge engineers involved in ontology research focus their work on *tools* to build ontologies and *ontological formats* to represent ontologies. Although both topics are of utmost importance their center of attention is of no direct interest for the growing numbers of scientists and engineers building ontologies. Tools and formats will therefore be discussed here only briefly.

3.5.2 Ontological formats

The choice of the ontological format has consequences for what can be represented in that format.
But formats are – in my opinion – of restricted importance, because proper ontology tools allow switching from one format to the other. On the other hand each format has its own representation power and switching from one format to another can lead to a loss of information, if the new format cannot represent (some) features of previous format.

[9] This GMP Handbook is also discussed in Chapters 1, 2 and 6.
[10] Tom Gruber said: *The term "Semiformal Ontology" refers to an ontology which has a few bits of formality but is largely informal. It is the analog of what Tom Malone calls semistructured data, such as email or office forms. A semiformal ontology could support technology to processing of its formal parts but leaves it to the reader make sense of the informal parts.* (in Lytras, M.D. and T. Gruber, 2004. Every ontology is a treaty - a social agreement - among people with some common motive in sharing, Official Quarterly Bulletin of AIS Special Interest Group on Semantic Web and Information Systems, 3, pp. 1-5.)
[11] An ontological format can be defined as *pattern into which knowledge is systematically arranged for use on a computer.*

A self-defined format may be sufficient for the demands set by the representation power needed, but in that case one has to define everything that is implicitly available in predefined, of-the-shelf ontological formats. These provide all basic terminology needed in all ontologies, which otherwise has to be defined before it can be used.

Well known ontological formats from the early years of ontology research, when ontologies were built on monolithic servers like the one at Stanford University Knowledge Systems Laboratory, are Ontolingua (Gruber, 1993) and KIF, i.e. Knowledge Interchange Format (Ginsberg, 1991, Genesereth and Fikes, 1992). At present many ontologies are built on stand-alone PCs or network servers. More recent formats for PCs and servers include XOL (XML-Based Ontology Exchange Language, Karp, 1999), RDF(S) (Resource Description Framework or RDF Schema[12]), OIL (Ontology Inference Layer[13]), DAML+OIL (DARPA Agent Markup Language[14] in combination with OIL). Furthermore, there are ontological formats associated with ontology development tools, support of OKBC[15] and JDBC[16] Databases. OWL or Web Ontology Language is the successor of RDF and RDF-S and is based on OIL and DAML+OIL, and is at present the preferred recommendation of W3C[17].

3.5.3 Ontological tools

Next to the selection of a proper ontological format, ontology developers need a tool. In the early years of ontology development of ontologies was supported by a central server application, e.g. the Ontolingua Ontology Editor (Farquhar *et al*., 1996). At present there are many tools, often developed in Java and therefore operating system independent. Some of these tools that have to be mentioned here are OntoEdit, OilEd and Protégé. The latter tool combines user-friendliness with an extended functionality.

Protégé is rather stable, as it exists since more than 15 years now. It has a large user group of helpful users and developers and it is open source software (written in platform independent Java with the source code available), allowing other developers to plug in their own extensions. Because of these features Protégé is very popular with more than 65,000 registered users worldwide (Noy, *et al*., 2000, Knublauch, 2003).

Protégé enables defining an ontology structure, it automatically generates forms facilitating knowledge acquisition and it allows building a knowledge base with its instance features and its import and exporting possibilities. Furthermore, Protégé is free, *open source* software and allows building application specific plug-ins for any missing functionality. Both other tools, OntoEdit and OilEd, offer powerful ontological formats, but lack the comprehensive and user-friendly features of Protégé.

Protégé can also be used in combination with a formal methodology for knowledge base development and management, e.g. CommonKADS (Schreiber et al., 2000). CommonKADS has been briefly discussed in Chapter 2, but it will not be applied here.

The present (full) version of Protégé[18] consists of three main parts: Core Protégé, Protégé-Frames, and Protégé-OWL. The Core Protégé is needed in combination with one of the other two parts, allowing to work either with the frame-based version or with the OWL-based version The OWL-based version is especially developed for editing ontologies for the Semantic Web and uses OWL as format. The frame-based version has an internal CLIPS (C Language Integrated Production System) format (Protégé's standard format), but also supports import and export of many formats, including XML, XML Schema, RDF, OIL DAML+OIL, UML and others.

[12] See http://www.w3.org/RDF/.
[13] See http://www.ontoknowledge.org/oil/.
[14] See http://www.daml.org/.
[15] See http://www.ai.sri.com/~okbc/.
[16] See http://protege.stanford.edu/doc/design/jdbc_backend.html.
[17] The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (W3).
[18] Protégé release version 3.2.1 (accessed on April 1, 2007).

Because of the reasons outlined here (user friendliness, stability, popularity and superior functionality) *Protégé*[19] has been chosen to develop the modelling knowledge base, the core part of the proposed framework to support multidisciplinary model-based problem solving. But even this easy and user-friendly tool appeared to be an obstacle for knowledge domain experts that were not used to applying knowledge engineering tools in their daily practice. Therefore a web based front-end knowledge base editor has been built that allows domain knowledge experts without knowledge engineering skills to view and edit the contents of the knowledge base (Kassahun *et al*., 2004).

The choice of a format and a tool has also other consequences. Tools often use their own slang for concepts and relations. In Protégé concepts are called *classes* and *relations* are defined as *slots*, but these *slots* are also used to specify *properties* of *classes*. *Classes* may have *subclasses*, which inherit *properties* of their *parent class*. If we define the *class* WINE, *subclasses* may be RED WINE and WHITE WINE. In case of such *subclasses* the relation between the *parent class* (WINE) and the *subclasses* (RED WINE and WHITE WINE) is of the type ISA (i.e. 'is a'). Basic *slots* as ISA do not need to be defined as these belong to a predefined *class* (in Protégé called SYSTEM CLASS, containing STANDARD CLASSES and STANDARD SLOTS, both essential to define ontologies as they contain many concepts at a very basic or meta-level. These do not have to be defined by Protégé users and may be used directly in the development of an ontology.

In conclusion, the choice of an ontological format and a tool for building ontologies is not crucial, as long as this choice does not hinder the team to switch to a new and better format or tool.

3.6 Ontologies in multi-disciplinary model-based problem solving

As stated before, ontologies can be used for representing different types of knowledge. Chandrasekaran *et al*. (1998, 1999) distinguish *domain factual knowledge* and *problem solving knowledge*. In the remainder of this book, both types of knowledge will be represented in ontologies.

The framework to support multidisciplinary model-based problem solving proposed in this book aims at structuring knowledge about the process of defining and supporting *modelling*, the specification of *problems* (with its *objects systems*) and the definition of appropriate *models*. Knowledge about the process of defining and supporting *modelling* belongs to the category of *problem solving knowledge*. Knowledge about the specification of *problems* (with its *objects systems*) and the definition of appropriate *models* is *domain factual knowledge*. To bootstrap these three ontologies (modelling, problem and appropriate models) a fourth ontology has been developed and will be discussed in Chapter 5. Following Uschold *et al.* (1998) the meta-ontology has to provide basic terminology, concepts and relations for the three other ontologies.

3.7 References

Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. University of Twente, The Netherlands, Enschede.

Borst, W. N., Akkermans, J. M., and Top, J. L. 1997. Engineering Ontologies. International Journal of Human-Computer Studies, 46: 365-406.

Chandrasekaran, B., J. Josephson, and V. Benjamins. 1998. Ontology of tasks and methods. in Eleventh Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada.

Chandrasekaran, B., Josephson, J., Benjamins, V. (1999). What are ontologies and why do we need them? IEEE Intelligent Systems 14, 20-26.

Date, C. (1975). An introduction to database systems. Addison-Wesley, Reading.

de Clercq, P. A., A. Hasman, J. A. Blom, and H. H. M. Korsten (2001). The application of ontologies and problem solving methods for the development of shareable guidelines. Artificial Intelligence in Medicine 22:1-22.

Farquhar, A., R. Fikes and J. Rice, 1996. The Ontolingua server: A tool for collaborative ontology construction, Knowledge Systems Laboratory, Stanford, Technical report, Stanford KSL 96-26, 19 pp.

Fox, M.S., (1992), The TOVE Project: A Common-sense Model of the Enterprise, In: Belli, F. and Radermacher, F.J. (Eds.), Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Artificial Intelligence # 604, Berlin: Springer-Verlag, pp. 25-34.

19 See for Protégé also the Protégé Project, http://protege.stanford.edu (access March 2007).

Genesereth, M. and R. Fikes, 1992. Knowledge Interchange Format, Version 3.0 Reference Manual;, Computer Science Department, Stanford University, Report Technical Report Logic-92-1, 1992. Also, KSL Technical Report 92-86 (http://www.ksl.stanford.edu/KSL_Abstracts/KSL-92-86.html).

Gennari, J.H., M.A. Musen, R.W. Fergerson, W.E. Gfrosso, M. Crubézy, H. Eriksson, N.F. Noy and S.W. Tu, 2002. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development, Report SMI Report Number: SMI-2002-0943, 32 pp. http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2002-0943.html.

Ginsberg, M.L., 1991. Knowledge Interchange Format: the KIF of death. AI Magazine 12, 57-63.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition 5((2)): 199-220.

Gruber, T. R. (1995). Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43:907-928.

Gruninger, M., and J. Lee. 2002. Ontology, applications and design. Communications of the ACM 45:39-41.

Houba, I.H.G., R.J.M. Hartog, J.L. Top,. A.J.M. Beulens and L.N. Berkel (2000). Using recipe classes for supporting detailed planning in food industry: A case Study. European Journal of Operational Research, 122, 367-373.

Karp, P.D., V.K. Chaudri and J. Themere, 1999. XOL: An XML-based Ontology exchange Language, Report. http://www.ai.sri.com/pkarp/xol/.

Kassahun, A., H. Scholten, G. Zompanakis, and C. Gavardinas. 2004. Support for model based water management with the HarmoniQuA toolbox. in C. Pahl, S. Schmidt, and T. Jakeman, editors. iEMSs 2004 International Congress: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004.

Klir, G.J., 1985. The Architecture of Systems Problem Solving, Plenum Press, New York.

Knublauch, H., 2003. An AI tool for the real world: Knowledge modeling with Protégé. JavaWorld, Access 20 September 2003, http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege.html.

Noy, N.F., R.W. Fergerson and M.A. Musen, 2000, The knowledge model of Protege-2000: Combining interoperability and flexibility, 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-2000-0830.pdf.

Scholten, H., A. Kassahun, J. C. Refsgaard, T. Kargas, G. Zompanakis, C. Gavardinas, and A. J. M. Beulens. 2005. The HarmoniQuA methodology to support multidisciplinary model-based water management. Environmental Modelling and Software submitted.

Scholten, H., R. H. Van Waveren, S. Groot, F. Van Geer, H. Wösten, R. D. Koeze, and J. J. Noort (2000). Good Modelling Practice in water management. in Proceedings HydroInformatics2000. International Association for Hydraulic Research, Cedar Rapids, Iowa, USA.

Scholten, H., R. H. Van Waveren, S. Groot, F. Van Geer, J. H. M. Wösten, R. D. Koeze, and J. J. Noort (2001). Improving the quality of model-based decision support: Good Modelling Practice in water management. Pages 223-230 in A. Schumann, J. Xia, M. Marino, and D. Rosbjerg, editors. Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the International Association of Hydrological Sciences, Maastricht, The Netherlands, July 18-27, 2001. IAHS, Maastricht, The Netherlands.

Schreiber, G., M. Crubézy and M. Musen, 2000. A case study in using Protégé-2000 as a tool for CommonKADS. In: R. Dieng, O. Corby (Eds.), Knowledge engineering and knowledge management: methods, model and tools, Lecture Notes in Artificial Intelligence, subseries of Lecture Notes in Computer Science, refereed proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2000, held in Juan-les-Pins, France in October 2000., J.G. Carbonell, J. Siekman, vol. 1937, Springer, Berlin, Heidelberg, New York, Barcelona, Hong Knong, London, Milan, Paris, Singapore, Tokyo, ISBN 3-540-41119-4.

Uschold, M, King, M, Moralee, S. and Y. Zorgios (1998) The Enterprise Ontology. The Knowledge Engineering Review , Vol. 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate).

Uschold, M. and M. Grüninger (1996) Ontologies: Principles, methods and applications, Knowledge Engineering Review, 11 (2): 93-155.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze, J. Noort. (1999). Good Modelling Practice Handbook, STOWA, Utrecht, RWS-RIZA, Lelystad, The Netherlands (in Dutch; English version from http://informatics.wur.nl/research%20projects/pub-pdf/gmp.pdf).

4 Multidisciplinary model-based problem solving

"The man who is striving to solve a problem
defined by existing knowledge and technique
is not just looking around.
He knows what he wants to achieve,
and he designs his instruments and directs his thoughts accordingly."
(Thomas Samuel Kuhn, 1922-1996)

Content of Chapter 4

4.1 Why this chapter?

A growing number of organizations in industry, service providers, banking and insurance, primary production, public administration, business consultants and knowledge institutes acknowledge the need of models and actually use them to cope with their complex problems. Model-based problem solving often requires a comprehensive analysis of a model that adequately represents relations between possible decisions and expected results of their implementation. Each problem requires a particular model[1] (or a set of models). Not all problems belong to a single (scientific) discipline, but many are multidisciplinary[2]. Multidisciplinary model-based problem solving teams are usually composed of experts originating from different disciplines, each with their own expertise, their own terminology, their own practices and methodology, sharing the more general parts of their know-how (e.g. the scientific method), but often differing in background, which hinders easy co-operation within the teams.

Knowledge plays an important role in this kind of problem solving. Knowledge can be ordered and structured in more or less formal ways, implicitly or explicitly. This chapter aims to structure knowledge related to model-based problem solving as explicitly as possible in three main groups: (1) knowledge about real-world problems[3], (2) knowledge related to models, i.e. representations of the real world in one or more models and (3) knowledge about the process[4] of representing and modelling. Knowledge technologists prefer to structure knowledge quite close to a problem/solution and implementation of the solution. Here a more open approach will be followed in order to facilitate re-use of (parts of) the knowledge and get a more impartial approach.

This book aims at supporting multidisciplinary model-based problem solving. First, differences between monodisciplinary and multidisciplinary problem solving have to be identified. Subsequently, problems related to these differences have to be recognized and an attempt made solve them. Teams consisting of members from various disciplines have to communicate and cooperate with each other. Because of their different backgrounds communication and co-operation is difficult. In this chapter I will explain the problems and subsequently propose a knowledge artefact, which allows defining a vocabulary for multidisciplinary model-based problem solving. On top of this artefact a framework will be proposed that structures problem related knowledge, model content knowledge and knowledge about the modelling process, all prerequisites for supporting model-based problem solving.

In the subsequent section a detour to philosophy of science will supply some vocabulary to extend monodisciplinary model-based problem solving to multidisciplinary model-based problem solving. A discussion of a model-based problem solving methodology[5] requires a vocabulary, which will be derived from philosophy of science, especially from the work of Thomas Kuhn, who introduced the concept ‘paradigm’. In the next section I will extend the science philosophical approach for monodisciplinary science to a multidisciplinary approach. I will outline an ontological way to structure knowledge. Such an ontological approach helps organizing diverse and unstructured knowledge. It allows applying this

---

[1] Many commonly used *models* can be classified as *mathematical models*, a set of relations (such as equations or inequalities) between model variables representing quantitative inputs (including *decisions*) and outputs (including *performance indices*) that measure consequences of implementation of decisions. Many of these *mathematical models* are process models, describing a set of processes, which can be distinguished in the real world or in an object system (i.e. that part of the real world that contains relevant processes for the problem at hand). Further on the term *model* stands for *mathematical model* unless specified otherwise (see Chapter 2, section 2.3).

[2] A *multidisciplinary* study combines disciplinary components, but integrates these components to a lesser extent than an *interdisciplinary* study. *Interdisciplinary* studies have their own theoretical, conceptual and methodological identity, which make them more coherent and integrated (Besselaar and Heimerik, 2001).

[3] In Chapter 5 *real-world problem* is defined as *a problem which is not a research question and not an exemplar, but refers to a complex problem, as can be found the context of the real world.* Furthermore, *real-world problems* should not be confused with *exemplars.* Although research questions are complex they are not synonyms of *real world problems*.

[4] *Process* means here the process of modelling (including representing the relevant aspects in a model). This should not be confused with physical, chemical, biological, socio-economical or business processes that are or have to be included in a model.

[5] *Methodology* has two different meanings (WordNet Dictionary): (1) the system of methods followed in a particular discipline; (2) the branch of philosophy that analyzes the principles and procedures of inquiry in a particular discipline. In this book I will always use *methodology* in the sense of the first meaning.

knowledge in all kinds of applications is flexible and enables re-use of (parts of) the knowledge. In the last sections of this chapter I come back to multidisciplinary model-based problem solving. Finally, a meta-ontology will be proposed to facilitate discussing and developing ontologies for multidisciplinary modelling, problems and their object systems and models to help solving the problems.

4.2 The scientific methodology cf. Thomas Kuhn

Thomas S. Kuhn proposes in his 'The Structure of Scientific Revolutions' a view on developments in scientific disciplines (Kuhn, 1970). He posits that scientific progress is not a continuous process, but occurs in revolutions as a result of a 'paradigm shift'. A central concept in his major work is the term 'paradigm'. In his later work he replaces this term by a self-defined (and therefore less biased) term 'disciplinary matrix'. Some of his readers have revealed that he did not use the word paradigm consistently, but with many (more than 20) different meanings. I will substantiate the term paradigm in this section. In subsequent sections I will use the concept paradigm and relate 'paradigm' to concepts like 'problem', 'problem solving method', 'model', and '(the) modelling process'. Further I will use ontologies to structure these concepts.

Kuhn needed the concept 'paradigm' in order to describe revolutions in science, not to describe a static situation of a scientific discipline. In his landmark contribution to philosophy of science (Kuhn, 1970) he did not intend to develop a vocabulary to describe scientific practice as is needed and is used to solve real life problems. His view on science was focused on describing the progress in science as a discontinuous phenomenon, driven by revolutions. Carl Popper (1959) described developments in science as a more continuous process, based on corroboration of theories and hypotheses. Each test, which cannot falsify a theory or hypotheses, increases the affirmation and belief in it. In the opinion of Lakatos (1978) scientific practice has a strong social component, which influences and determines scientific progress to some extent. In my opinion these three figureheads of the philosophy of science have contributed significantly to the present view on scientific practice and its progress. A detailed discussion of scientific practice is out of scope here, as I only need some vocabulary to describe and discuss mono- and multidisciplinary science and a structure for knowledge underlying science applied to real world problems.

Kuhn's term 'paradigm' is used in many different ways. Therefore I have to be explicit in the way I use it here. Paradigm refers to a scientific community consisting of scientists sharing a discipline in their education and in their profession. A basic term, used in this book and especially in this chapter is 'problem', which will be used here with the extended meaning: *a situation asking for understanding or a wanted solution*[6] *using scientific methods*. Rephrasing Kuhn's meaning of the concept paradigm, a paradigm consists of the following elements, for which I propose the term 'paradigm describing aspects'. I define the concept 'paradigm' with the following paradigm describing aspects:

- **Homogeneous scientific community**: a group of scientists with comparable (academic) education, working in the same field (although often with different roles, e.g. client, problem solver, researcher), meeting each other at the same conferences, publishing in the same journals.
- **Shared worldview** (*Weltanschauung*): based on their shared educational background and based on sharing all the other paradigm describing aspects; their shared worldview consists of non-scientific concepts, including non-scientific socio-economic aspects and general scientific concepts, which not strictly belong to the paradigm.
- **Shared perceptions, concepts and terminology**: a set of perceptions, concepts and terms, shared by the group; this set is often inconsistent in details, but it allows members of the group to understand each other scientifically.
- **Accepted theories and hypotheses**: a set of consistent theories, thoroughly tested and published, accompanied by less tested hypotheses, which did not reach the status of theories yet; these theories form the heart of a paradigm together with the *methods*; any fact that does not fit in the existing set of accepted theories, will be molded until it fits or until a new paradigm is proposed and is supported by a group of scientists (the *homogeneous scientific community* of the new paradigm).
- **Shared general assumptions**: a set of generic assumptions, i.e. not accounting for non-scientific social aspects, which are included in *worldview*, but assumptions required for the consistency of the *theories* and *methods* and therefore boundary conditions for a paradigm.

[6] It can be necessary to remove an obstacle or to achieve a more wanted situation.

- **Shared practices**: the set of practices needed to apply *methods* to solve problems that fit directly into the *exemplar*[7] straitjacket or that can be solved within the paradigm after some adaptations.
- **Accepted methodology**: the set of accepted methods are the result of a process consisting of proposing and testing methods according to the shared scientific practice, often applied on problems that are similar to *exemplars* or that can be molded in the form *exemplars* have; these methods are discussed on conferences visited by the group and published in journals/textbooks by the group; together with the accepted theories it forms the heart of a paradigm.
- **Exemplars**: rather simple problems, which fit perfectly in the paradigm, which can be solved with the methods of the paradigm and which solutions do not contradict with theoretical foundations of the paradigm; exemplars can best be seen as textbook problems and certainly not as real world problems.
- **Exemplar specific assumptions**: a set of (detailed) assumptions that transform and simplify a real problem into an exemplar.

In summary, within a paradigm, scientists, professionals and other practitioners share their educational background, theories, methods and practices, resulting in their own esoteric language for communication within the paradigm (including books, conference presentations and journal papers). The 'paradigm describing aspects' are structured in Figure 4-1. The concept 'paradigm' refers to what Kuhn calls 'normal science', i.e. monodisciplinary science or in other words a branch of science in which the scientific community is homogenous with a similar training in a single scientific discipline[8] (e.g. hydrologists, ecologists, environmental scientists, etc.).

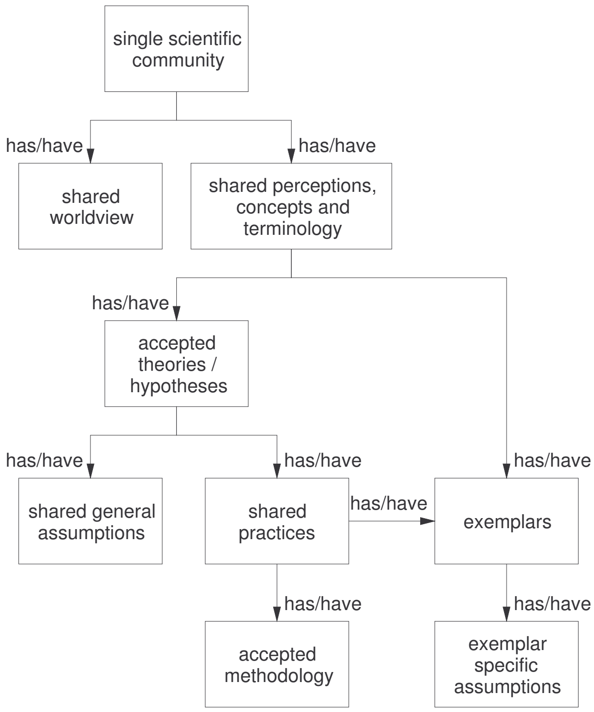


Figure 4-1. An ontological structure for the paradigm describing aspects of Kuhn's 'normal' science. Rectangles are concepts and arrows are relations[9]. Ontologies were introduced in Chapter 3.

4.3 A paradigmatic view on multidisciplinary science

The description of paradigms, summarized in the previous section is adequate and sufficient to explain scientific developments, but it refers to 'normal' science (cf. Kuhn), which is typically monodisciplinary (by

[7] *Exemplar* is a term of Thomas Kuhn, meaning *example* in textbooks within the paradigm; see also http://en.wikipedia.org/wiki/Exemplar.

[8] The WordNet dictionary refers to '*discipline*' as a branch of knowledge; I will use '*discipline*' and '*paradigm*' as synonyms, but using '*paradigm*' emphasises the science-philosophical context and '*discipline*' the more practical context of problem solving projects and the background of team members in such projects. '*Domain*' is also used as a synonym, emphasizing the problem context.

[9] Arrows have to be interpreted as sentences with the following grammar: {source of arrow = subject | arrow name = part of speech, usually an action, occurrence or state of being | destination of arrow =object}.

definition, if paradigm and discipline cover the same concept). As said in section 4.1, at present many problems are complex and require a comprehensive analysis with one or more models, which are often of a multidisciplinary character. Members of model-based problem solving teams have therefore different backgrounds, which hinder communication and cooperation. In this section I will attempt to identify paradigmatic differences between mono- and multidisciplinary science, specified in the paradigm describing aspects discussed earlier on. Science should be interpreted here as a problem solving process, and thus as applied science.

The structure of a multidisciplinary paradigm is equal to the structure of a monodisciplinary paradigm (see Figure 4-2 and compare it with Figure 4-1). Differences in the paradigm describing aspects can be summarized as follows:

- **Multidisciplinary team**: instead of a homogeneous scientific community a team of experts with different scientific expertise (education, experience, profession) co-operate to solve real problems; they meet each other at workshops, but visit (partly) disjoint scientific conferences and publish in different journals; multi-disciplinary teams usually exist on the base of a project to solve 'real problems' and not at an institutional level.
- **Partially overlapping[10] worldviews**: although worldviews vary between different disciplines/paradigms, a common educational base (the common part in science education) often facilitates co-operation in a multidisciplinary project.
- **Partially disjoint[11] perceptions, concepts and terminology**: as each discipline/paradigm has a set of perceptions, concepts and terms; a multidisciplinary team requires a substantial effort to overcome differences between sets of perceptions, concepts and terminology belonging to different disciplines/paradigms to enable mutual understanding of team members and fruitful co-operation.
- **Partially disjoint theories and hypotheses**: in the multidisciplinary case several sets of theories and hypotheses exist, which are consistent within the set, but which may be inconsistent between sets; multidisciplinary projects hardly expend resources to identifying these inconsistencies, and even less to eliminate them.
- **Partially overlapping general assumptions**: although some of these assumptions, necessary for the theories and hypotheses, coincide, they are directly related to the partly disjoint sets of theories and hypotheses.

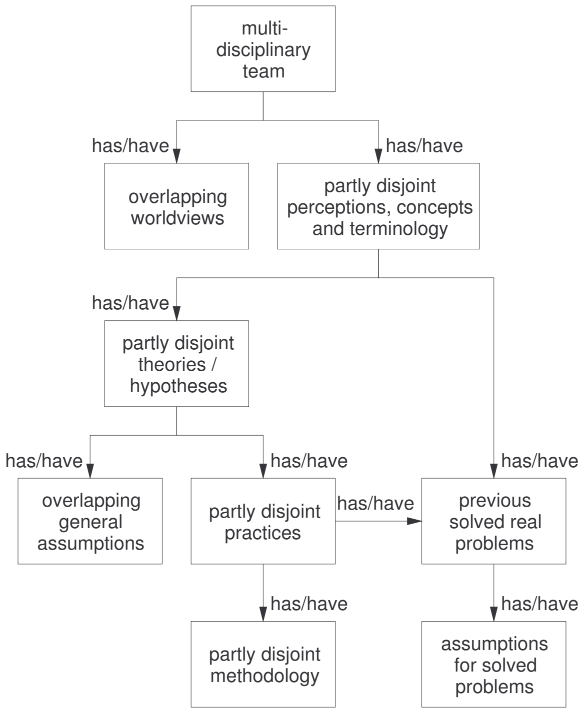


Figure 4-2. An ontological structure for the paradigm describing aspects for multidisciplinary paradigms (opposite to Kuhn's 'normal' science). Rectangles are concepts and arrows are relations. Ontologies are introduced in Chapter 3.

[10] *Partly overlapping* accentuates the overlap and lessens the significance of differences.
[11] *Partly disjoint* accentuates the differences and lessens the importance of the overlap.

- **Partially disjoint practices**: if 'real problems' are solved in multidisciplinary projects, each discipline/paradigm will usually have its own practices; these sets of practices partly overlap, but the rest will vary between disciplines/paradigms. Within the project different practices should be handled – preferably – in a formal way by using explicitly described ways how to deal with incompatible practices.
- **Partially disjoint methodology**: in the multidisciplinary case several sets of methods, directly related to the various disciplines/paradigms; the methodology, available within the team in a multidisciplinary project, does not automatically fit together and it may require considerable resources of the project to cope with these kind of obstacles; incompatibilities and approaches on how to deal with these are hardly discussed at conferences.
- **Previously solved real problems**: since real problems are assessed at present in a multidisciplinary way, a growing number of cases will come available that can be used as (to some extent formalized) examples of how to overcome complications caused by the multidisciplinary character of the problem; these are not *exemplars*, as multidisciplinary new projects (1) are too complicated to be seen as *exemplars* and (2) belong generally to incomparable assemblages of disciplines; multidisciplinary teams should not try to fit new problems in the straitjacket of previously solved real problems.
- **Assumptions for solved problems**: although usually the assumptions for *previously solved real problems* will not be adequate for new real problems; lists of assumptions of comparable, *previously solved real problems* may act as starting point for a set of assumptions for new real problems.

Comparing the *paradigm describing aspects* of a monodisciplinary paradigm (a paradigm for 'normal science' cf. Kuhn) with those of a multidisciplinary model-based problem solving paradigm shows that many new difficulties arise. The four major difficulties due to a multidisciplinary character of problems can be summarized as follows:
1. *Partially* disjoint *perceptions, concepts and terminology* will endanger optimal communication between team members in multidisciplinary teams;
2. *Incompatibilities between theories and hypotheses*, originating from different disciplines/paradigms, constitute relatively concealed obstacles;
3. *Incompatible methods*, originating from different disciplines/paradigms, are problematic, if they have to be jointly applied;
4. *Incompatible practices* hinder multidisciplinary projects directly.

These problems, associated with the multidisciplinary character of problems, have to be dealt with, if one aims to support multidisciplinary model-based problem solving projects. Preferably these hurdles should be removed by solving these problems or at least all problems should be made explicit. A large part of this book is spent to practical support to help scientists and practitioners co-operate in multidisciplinary teams, not only by providing a methodology or setting up a methodological artifact, but also by providing guidance and tools for direct support of daily practice for all involved in multidisciplinary model-based problem solving.

The overall approach is derived from quality assurance initiatives in industry copied from software engineering (Humphrey, 1989) and converted to modelling and simulation (Scholten and Udink ten Cate, 1995, 1999)[12]. The heart of this approach is *defining* the complete process, in this case defining multidisciplinary model-based problem solving. The technology, which enables defining such a complex process, will be borrowed from knowledge engineering. More specifically, the process will be defined using *ontologies*. A short introduction on ontologies and how these can be applied to structure procedural knowledge as is required to define this complex process has been given in Chapter 3.

4.4 The multidisciplinary model-based problem solving process

Complex ('real') problems are often too complicated to be solved within a single model-based problem solving paradigm. Usually teams are composed of experts with different backgrounds, typically belonging to several paradigms, normally called multidisciplinary teams. The problem owner or his representative selects the members of such teams, based on his understanding of the problem at hand and his intuitive educated guess of the type of expertise required for solving the problem. In the past many problem solving teams have been composed of members belonging to a single paradigm, which easily lead to forcing the problem at hand within this paradigm and hindering a more effective multidisciplinary approach. At present multidisciplinary problem solving teams are more common. This has two crucial reasons: (1) firstly, many problems could not

[12] See also Chapter 2.

satisfactorily be solved within a single paradigm and (2) at present computers are more powerful and problem solving methodologies are more sophisticated. Problem owners today explicitly call for different aspects to be treated by integrating knowledge and methods originating from different paradigms.

A multidisciplinary approach introduces new hurdles in the problem solving process. These barriers are associated with fences between the different paradigms the team members belong to. Within each of these paradigms, Kuhn's view on paradigms and disciplines outlines a structure to solve problems, but as soon as problem solving team members originate from various paradigms, several hurdles have to be taken, which are outlined above in section 4.3. *Problems with different worldviews* are usually of no importance at all, as the worldviews of most scientists are not so different after all. More significant are *incompatible theories and hypotheses*. In many cases the theories and hypotheses are just complementary and in many of the remaining cases incompatibilities should be harmonized at some academic level and not within a problem solving project context. Only in case these incompatibilities obstruct the process of problem solving itself, serious action should be taken. The problem of *incompatible methods* can cause major problems. Until now there are no general solutions or software, which act as *panacea* to overcome this problems. Fortunately several initiatives are working towards (partial) solutions, some by enabling reliable, 'smart' interfaces between models from different paradigms (Gijsbers *et al.*, 2002, Blind and Gregersen, 2004, Moore *et al.*, 2004, Gijsbers, 2004, Westem *et al.*, 2004, Fortune, 2004), while others aim at developing a multiparadigm modelling infrastructure, which allows switching between methods originating from various paradigms and combining these methods in one multidisciplinary problem solving approach (Makowski, 2005). The remaining problems, i.e. *partially* disjoint *perceptions, concepts and terminology* and *incompatible practices* are less apparent and do not obstruct any process of multidisciplinary model-based problem solving directly, but they hinder an efficient and effective process. Here I will aim at lowering these hurdles concurrently by developing ontologies to harmonize *perceptions*, *concepts* and *terminology* within a multidisciplinary model-based problem solving project and to harmonize *practices* originating from different disciplines.

4.5 A first blueprint of multidisciplinary model-based problem solving

Now that the extra problems with multidisciplinary model-based problem solving have been identified and the approach has been chosen how to deal with these problems, a first blueprint of the model-based problem solving process can be presented:

- If some problem owner (a person or an organization) has a 'real problem', the problem owner will bring together a multidisciplinary team consisting of team members with the appropriate paradigmatic background for the job at hand, i.e. proper competency of the team. This choice strongly depends on:
    - The view of the problem owner on the problem;
    - The problem owner's intuitive blueprint on what should be done to solve the problem;
    - The paradigmatic/disciplinary background of the problem owner;
    - The previous experiences of the problem owner with projects that aim to solve similar or comparable problems.

  At this point there are a multidisciplinary team and a problem owner of a complex, real world problem, which is not yet precisely defined. As soon as the project starts, basically the following activities have to be undertaken (more details will be provided later; the activities are structured in Figure 4-3);
- A problem should be defined, based on (often vague) information about the real world (including the problem owners' mental model and the team members' mental models);
- The problem solving team identifies which problem solving paradigms are appropriate to solve the defined problem;
- These appropriate problem solving paradigms provide methods and practices belonging to a class of generic models;
- This class of generic models will be restricted by the real-world problem at hand to a set of generic, but adequate models;
- Aspects of the real world should be conceptualized into an object system[13], relevant for the problem at hand;
- The defined real-world-problem should be conceptualized to fit into one or more (coupled) adequate models, which should be selected from a class of generic models provided by available paradigms, resulting in a set of generically adequate models, belonging to one or more paradigms;

[13] In Chapter 5 *object system* will be defined as *the things and processes in the real world or some defined system that will be the object of modelling*.

- Coupling of generic, adequate models with the object system for the real-world-problem allows the specification of a particular model (which now depends on the object system and the defined real-world-problem);
- A set of scenarios[14] should be defined (depending on the problem, the OS, and the particular model);
- The set of scenarios should be applied to all models in the set of particular models with as result a set of evaluated scenarios;
- The best solution (multi-criteria analysis) should be selected from the generated ones;
- The best solution should be used to solve the problem;
- If necessary other models should be tested with the scenarios.

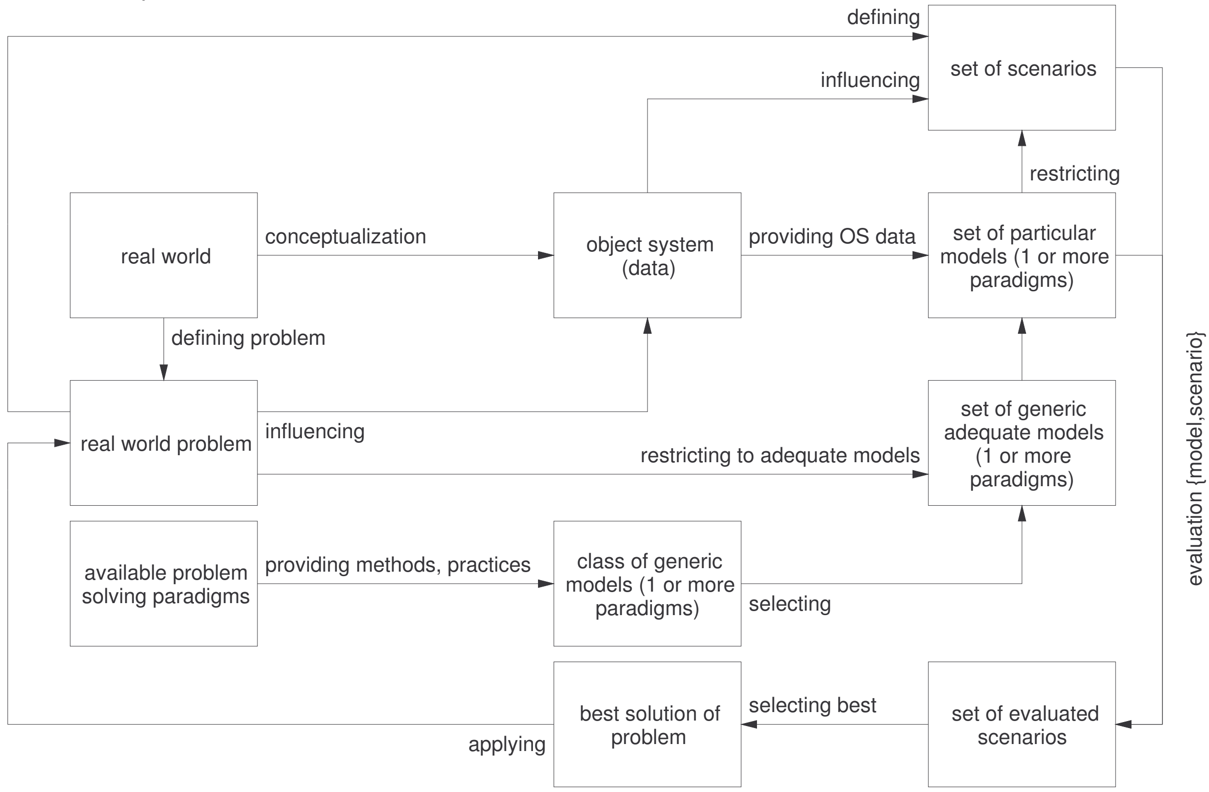


Figure 4-3. First outline of the multidisciplinary model-based problem solving process. Rectangles are ontological concepts and arrows are ontological relations, denoting actions/activities between the concepts. The actions/activities *defining the process* and the *multidisciplinary project team* have been left out intentionally.

The procedure outlined above will be discussed in more detail in subsequent Chapters. Here many details are left out, especially those related to the content of the model, i.e. the real-world processes to be modeled. Furthermore, some of the concepts and relations of the outlined procedure do not belong to the process of multidisciplinary model-based problem solving (i.e. modelling), but belong to the task of abstracting the problem from the real world and defining the Object System (OS). Finally, some parts of the outline presented in Figure 4-3, come from the ontologies for mono- and multidisciplinary problem solving (sections 4.2 and 4.3).

In order to introduce more structure in the blueprint of multidisciplinary model-based problem solving, presented in this section, I will describe two views. The first view aims at connecting the mono- and multidisciplinary paradigms from section 4.3 and section 4.4 with the blueprint of multidisciplinary model-based problem solving. The second view aims at designing a structure that can serve as an ontological framework, which can be detailed in the next Chapters. Both views represent a meta-model of modelling.

[14] The term ‘scenario’ can be used in two meanings: (1) chosen approach (e.g. in problem solving), (2) set of inputs or alternative choices (e.g. in OR or in simulation).

4.6 A first ontological structure for multidisciplinary model-based problem solving

To organize all components I propose a structure of **tiers**, in which multidisciplinary model-based problem solving fits in a single tier and all other components in a few other tiers. This structure can be summarized as follows (starting with the most basic tier and ending with the top tier, see also Figure 4-4):

- **Tier –3**: The ontology for the monodisciplinary model-based problem solving paradigm, i.e. Kuhn's 'normal' science (see section 4.2);
- **Tier –2**: The ontology for the multidisciplinary model-based problem solving paradigm (see section 4.3);
- **Tier –1**: The harmonization of the multidisciplinary model-based problem solving paradigm i.e. solving incompatibilities and inconsistencies due to integrating different paradigms (see the last paragraph of section 4.4);
- **Tier 0**: The actual processes related to multidisciplinary model-based problem solving occur in this tier (see the first part of section 4.5);
- **Tier 1**: Besides the processes of **tier 0**, many modelling processes aim at evaluation, checking, controlling and comparing (intermediate) results with higher-tier goals and expectations. Some examples are:
  - Evaluation of model selection, which compares the *representation need*, determined by the problem at hand and its definition with the *representation power*, determined by the set of paradigms provided by the paradigmatic background of the multidisciplinary team;
  - A qualitative evaluation of the particular model instance (does it allow to solve the problem at all?); some authors (e.g. Sargent, 1982, 1984a, 1984b, 1986) refer to this evaluation as *operational validation*;
  - A quantitative evaluation of the solutions (by comparing the scenarios run by the particular model);

  The processes in this **tier 1** can be labeled as 'evaluation of Good Modelling Practices' (see Chapter 6).
- **Tier 2**: Auditing, which is the process in the highest tier, controlling **tier 0** by using and checking **tier 1** (see the next Chapter).

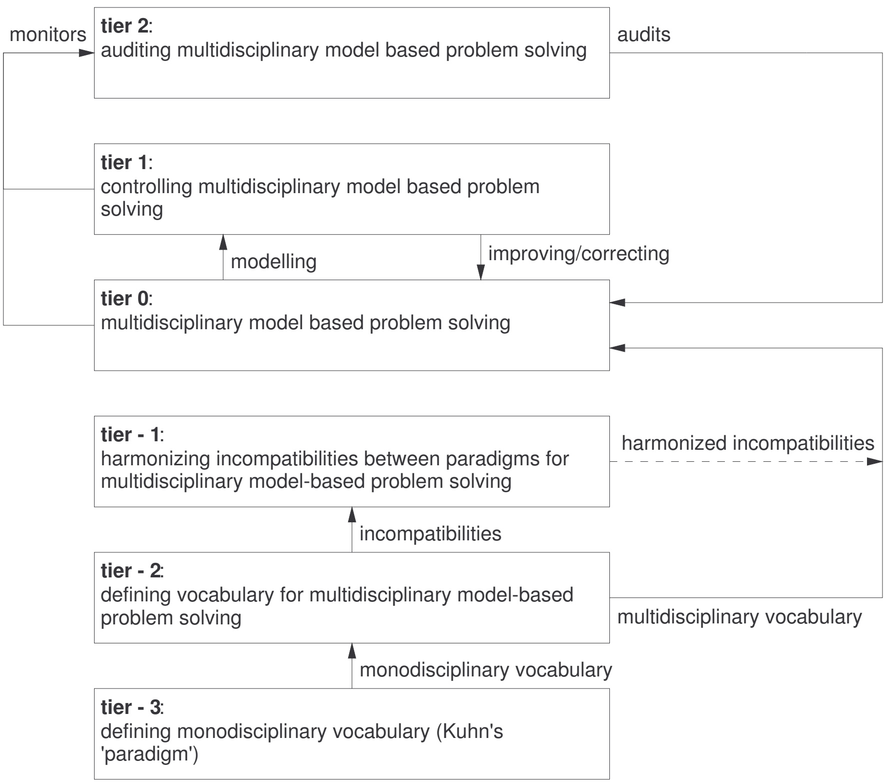


Figure 4-4. An ontological structure for the multidisciplinary model-based problems solving process. Details of **tier -3**, of **tier -2** and of **tier -1** are given in Figure 4-1, Figure 4-2 and sections 4.3 and 4.4. Actual modelling occurs in **tier 0** and evaluation, correcting and improving modelling occurs in **tier 1**. The **tier 0**, **tier 1** and **tier 2** will be discussed in detail in a subsequent Chapter.

To go from here to a more practical structure that can serve as a backbone for an ontological framework in which all aspects of model-based problem solving can be defined, I will combine **Tier –3**, **Tier –2**, and **Tier –1** into a meta-ontology for model-based problem solving (in short the meta-ontology) and I will refer to **Tier 0**, **Tier 1**, and **Tier 2** as the multidisciplinary model based problem solving ontology (in short the modelling ontology). In the next section, I will first endeavor to extend the latter two ontologies with two others, a model ontology and an object system ontology.

4.7 A second ontological structure for multidisciplinary model-based problem solving

In this chapter I have briefly introduced a structure and some vocabulary for multidisciplinary model-based problem solving. This ontological structure organizes all tasks and activities of the model-based problem solving process in **tiers** with the lower **tiers** (-3, -2, -1) providing vocabulary and building blocks to overcome problems caused by the multidisciplinary character. I named the ontology of these three lower tiers the *meta-ontology* for multidisciplinary model-based problem solving. The central **tier 0** (actual modelling) and **tier 1** (modelling control) and **tier 2** (modelling audit) will be called the multidisciplinary model-based problem solving ontology or *modelling ontology*. This structure covers only modelling and does not include what is relevant in the object system or the content of models that try to solve the problem at hand in the object system.

To solve some of the problems caused by the multidisciplinary character, the definition and use of two more ontologies might be instrumental to arrive at efficient and effective model-based problem solving with assured quality. In my opinion multidisciplinary model-based problem solving needs four ontologies (see Figure 4-5):

- A **meta-ontology:**
  - Provides vocabulary for the other three ontologies;
  - Uses the concepts and relations of Figure 4-1 and Figure 4-2;
  - Is depicted in Figure 4-4 and structured in **tier -3**, **tier -2** and **tier -1**;
  - Is discussed in Chapter 5;
- A **modelling ontology:**
  - Describes the activities of modelling;
  - Is depicted in Figure 4-4 and structured in **tier 0**, **tier 1** and **tier 2**;
  - Is discussed in Chapters 6 and 7;
- A **problem and object system ontology:**
  - Describes (parts and aspects of) the real world, the problem at hand and the object system;
  - Has (at least) two instances:
    1. The present situation;
    2. A future unrealized, but predicted, situation that is wanted by the problem owner and other stakeholders;
  - Is discussed in Chapter 8.
- A **model ontology:**
  - Comprises the content of a model (often indicated as domain knowledge);
  - Is discussed in Chapter 9;

Between these ontologies several (highly abstract) relations can be distinguished:

- The *meta-ontology* provides vocabulary to the *modelling ontology*, the *problem and object system ontology* and the *model ontology*;
- The *modelling ontology* defines the work to be done in the *model ontology* and to some extent the work in the *problem and object system ontology*;
- The *model ontology* uses the *problem ontology* and the *modelling ontology*;
- The work in the *problem and object system ontology* is organized in the *modelling ontology* and provides process knowledge to the *model ontology*.

The structured ontology for multidisciplinary model-based problem solving, outlined in Figure 4-5, will be detailed in the following Chapters, where the *modelling ontology*, the *problem and object system ontology* and the *model ontology* will be introduced as formally as possible. In Chapter 3 ontologies have been introduced with two *ontological layers*[15]: the *structure layer* and the *instance layer*. In many cases more ontological layers give more expressive power to the assemblage of ontologies. *Ontological layer i* should provide all knowledge necessary to define, discuss and use the knowledge structured in *ontological layer i+1*. In this way, a structured framework of ontologies allows ontology developers (experts) to add coherent detailed content separated from other details and not mixed up with more generic knowledge that is located in a higher ontological layer. This is not a new idea, because ontologies have always been assumed to be combinable into more powerful ones, whether by combining ontologies 'in the same ontological layer' or in hierarchical structures as in the 'Standard Upper Ontology'[16] or similar initiatives (Cyc[17], core-ontology[18]).

In this book a similar approach has been followed with a hierarchical structured ontological framework. Figure 4-5 links the *modelling ontology* with the three other ontologies (*meta-ontology*, *problem and object-system ontology* and *model ontology*).

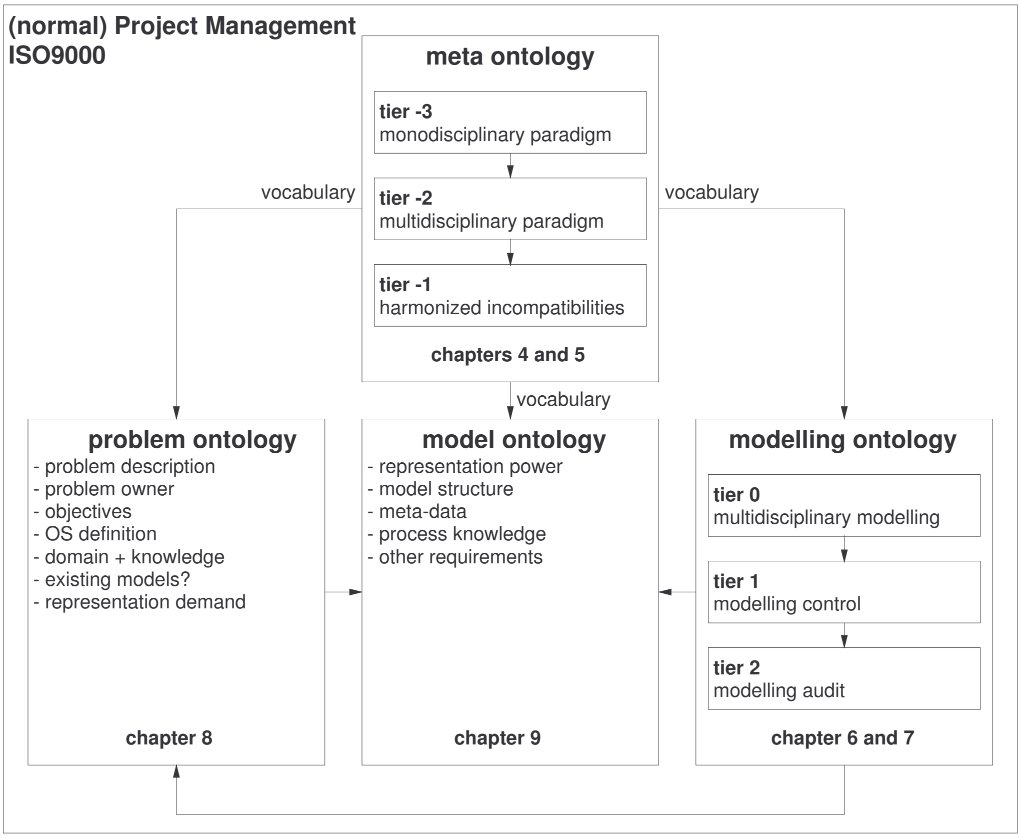


Figure 4-5. A structured ontology for multidisciplinary problem solving, consisting of 4 interrelated ontologies. At a highly abstract level, the ontologies are concepts (presented as rectangles) and (highly abstract) relations between the ontologies are depicted as arrows. The tiers and their numbering refer to the tiers shown in Figure 4-4.

The *modelling ontology* will be structured in ontological layers with increasing specialization. The highest ontological layer contains *generic process knowledge*, the next layer contains *modelling knowledge*, the following layer *simulation modelling knowledge* – most specialized layer – *modelling knowledge for water management*. The lowest layer (*modelling journal*) will be filled what a log of what modelling teams actually do in the process. Furthermore, each ontological element can branch out into one or more specializations, leading to a tree-like ontological framework (Figure 4-6), where *modelling knowledge*, *business process knowledge* and *socio-economic process knowledge* are depicted as alternatives. The next Chapter aims at

[15] *Ontological layer* refers to the level of specialization and is defined in the basic concepts of the meta-ontology (Chapter 5).
[16] There are many publications, but an up-to-date overview can best be found at http://suo.ieee.org.
[17] *OpenCyc* is the open source version of the Cyc technology, the world's largest and most complete knowledge base and reasoning engine, developed by Cycorp. An overview can be found at http://opensyc.org.
[18] A *core* ontology is a very basic, minimal, bootstrapping, ontology, consisting of the minimal concepts, required to develop other ontologies. The meta-ontology proposed in this book is a core-ontology for multidisciplinary problem solving.

focusing on the modelling thread resulting in a definition of the modelling process for water management and supports water management modellers throughout their work. Additionally, I will show that parts of the ontological structure for modelling can be reused for other purposes.

The *problem and object system ontology* and the *model ontology* will be defined in the Chapter 8 and Chapter 9. For these ontologies a similar approach will be followed as for the modelling ontology with a stepwise specialization ending in detailed instances filled with exemplary knowledge, selected because of my personal interests and experiences. The *problem and object system ontology* and the *model ontology* will be focused on knowledge from the field of bivalve ecophysiology, which aims at understanding the physiological response of shellfish to changes in their ecological environment. The *modelling ontology* will address the process of modelling for water management in general and details related to (sub)domains in water management, e.g. hydrodynamics, water quality, flood forecasting.

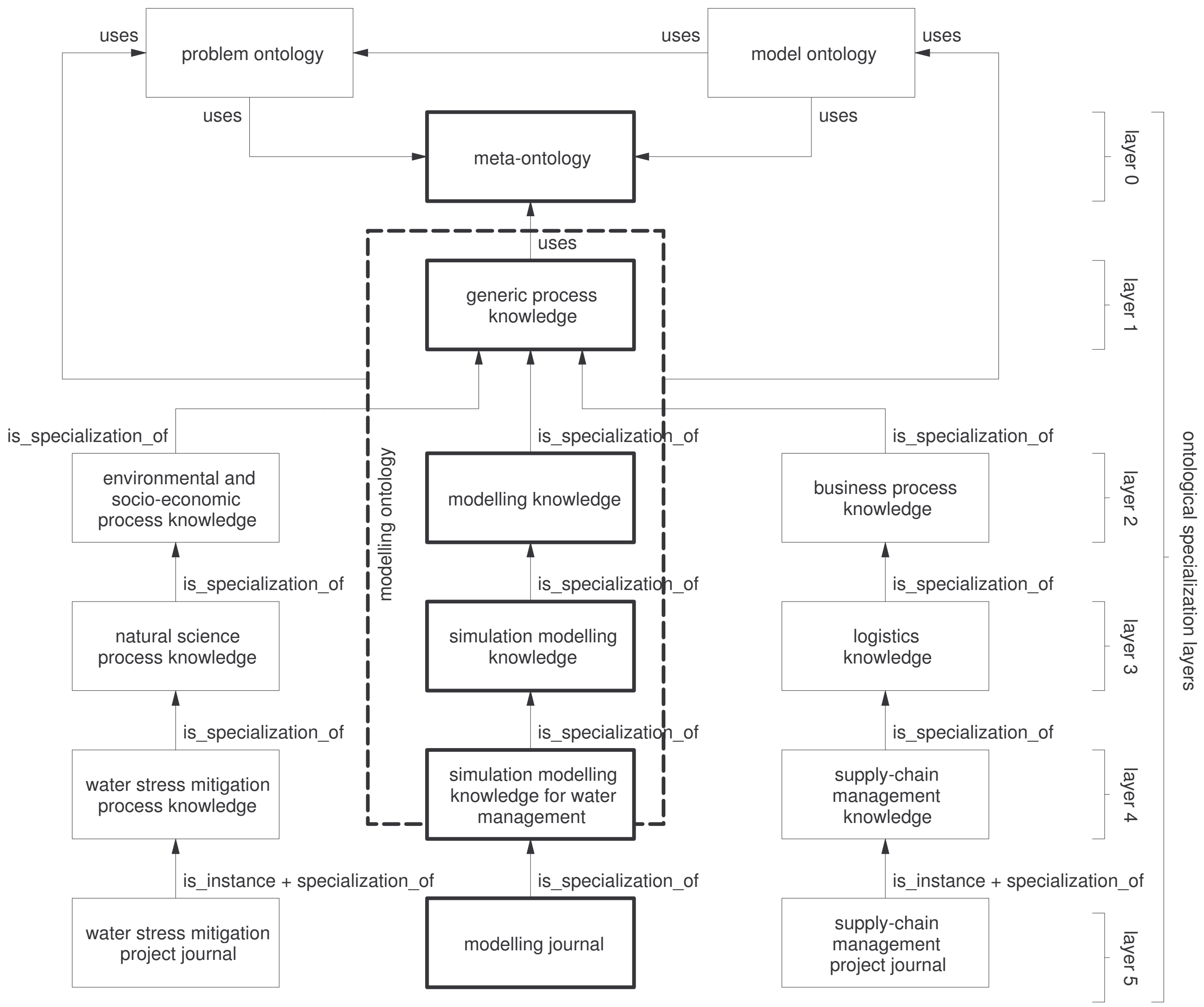


Figure 4-6. A stepwise ontology specialization with the process ontology, instantiated for modelling and expanded to some more particular ontological layers. The top (meta-ontology) is the most generic ontological layer and the concepts at the bottom the most specialized ones. The words *ontology* and *instance* are explained further in the text of this section and in more detail in Chapter 5. Bold lined concepts refer to the modelling thread, which will be detailed in the next Chapter.

4.8 References

van den Besselaar, P. and P. Heimeriks, 2001, Disciplinary, Multidisciplinary, Interdisciplinary: Concepts and Indicators. In: M. Davis, C.S. Wilson (Eds.), Proceedings 8th International Conference on Scientometrics and Informetrics - ISSI2001, Sydney, UNSW, 705-716, http://hcs.science.uva.nl/usr/peter/publications/2002issi.pdf.

Blind, M. W., and J. B. Gregersen. 2004. Towards an Open Modelling Interface (OpenMI). in C. Pahl, S. Schmidt, and T. Jakeman, editors. iEMSs 2004 International Congress: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany.

Fortune, D. 2004. Catchment Decision Support Systems using OpenMI. Pages 1835-1842 in Shie-Yui liong, Kok-Kwang Phoon, and V. Babovic, editors. Proc. of the 6th Int. Conf. on Hydroinformatics. World Scientific Publishing, Singapore.

Gijsbers, P. 2004. The OpenMI architecture - details. Pages 1819-1826 in Shie-Yui liong, Kok-Kwang Phoon, and V. Babovic, editors. Proc. of the 6th Int. Conf. on Hydroinformatics. World Scientific Publishing, Singapore.

Gijsbers, P., R. Moore, and C. Tindall. 2002. HarmonIT: towards OMI, an Open Modelling Interface and Environment to harmonise European developments in water related simulation software. Pages 1268-1275 in HydroInformatics 2002. International Association for Hydraulic Research, Cardiff, UK.

Humphrey W.S. (1989). Managing the software process, Addison-Wesley, Reading.

Kuhn, T., 1970. The structure of scientific revolutions. University of Chicago Press, Chicago, 210 pp.

Lakatos, 1978. The methodology of scientific research programmes. Cambridge University Press, Cambridge, 250 pp.

Makowski, M., 2005. A Structured Modeling Technology. European Journal of Operational Research 166, 615-648.

Moore, R., C. Tindall, and D. Fortune. 2004. Update on the HarmonIT Project - The OpenMI Standard for Model Linking. Pages 1811-1818 in Shie-Yui liong, Kok-Kwang Phoon, and V. Babovic, editors. Proc. of the 6th Int. Conf. on Hydroinformatics. World Scientific Publishing, Singapore.

Popper, K.R., 1959. The logic of scientific discovery. Unwin Hyman Ltd, London, 480 pp.

Sargent, R.G., 1982. Verification and validation of simulation models. In: F.E. Cellier (Editor), Progress in modelling and simulation, Academic Press, London, etc, 159-169.

Sargent, R.G., 1984a. A tutorial on verification and validation of simulation models. In: S. Sheppard, U. Pooch and D. Pegden (Editors), Proceedings of the 1984 winter simulation conference, pp 115-121.

Sargent, R.G., 1984b. Simulation model validation. In: T.I. Ören, B.P. Zeigler and M.S. Elzas (Editors), Simulation and model-based methodologies: an integrative view, 10 in the series: NATO ASI Series F: Computer and Systems Sciences, Springer Verlag, Berlin, etc., pp 537-555.

Sargent, R.G., 1986. An exploration of possibilities for expert aids in model validation. In: M.S. Elzas, T.I. Ören en B.P. Zeigler (Editors), Modelling and simulation methodology in the Artificial Intelligence era, Elsevier, Amsterdam, 19-40.

Scholten, H., and A. J. Udink ten Cate. 1995. Testing simulation modelling quality. Pages II.A.3.1-6 in Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium.

Westem, S., D. Fortune, and J. B. Gregersen. 2004. OpenMI - New Opportunities for Model Developers. Pages 1827-1834 in Shie-Yui liong, Kok-Kwang Phoon, and V. Babovic, editors. Proc. of the 6th Int. Conf. on Hydroinformatics. World Scientific Publishing, Singapore.

5 Bootstrapping an ontological framework for model-based problem solving

"Pooh looked at his two paws.
He knew that one of them was the right,
and he knew that when you had decided which one of them was the right,
then the other was the left,
but he never could remember how to begin"
(A.A. Milne, 1882-1956)

Content of Chapter 5

This chapter is based on parts of the following paper:

Bootstrapping ontologies

Beulens, A.J.M. and H. Scholten, 2005. Towards a process ontology for a model based support system for problem solving: the ontology bootstrap problem, M. Makowski, Knowledge Creation and Integration for Solving Complex Problems. The 19th International Workshop on Complex Systems Modeling (CSM) jointly with the 6th International Symposium on Knowledge and Systems Sciences (KSS), August 29-31, 2005, Laxenburg, Austria, 3-8. (*http://www.iiasa.ac.at/~marek/ftppub/Pubs/csm05/abst.pdf*).

5.1 Introduction

As discussed in Chapter 4, an ontological framework will be developed for multidisciplinary model-based problem solving, consisting of a *modelling ontology*, a *problem and object system ontology* and a *model ontology*. This framework will be built from scratch starting with a *meta-ontology* that provides vocabulary for the *modelling ontology*, the *model ontology* and the *modelling ontology*.

Chapter 4 (Figure 46) introduces '*ontological layer*s', being the degree of specialization. The modelling ontology is a part of the process branch. The modelling ontology is more specialized than the generic process ontology and more generic than the ontology for model-based water management. Each more specialized ontology can be seen as an instance of the previous, more generic one.

Ontologies are meant to represent relevant knowledge. Protégé, the ontology tool used, provides an ontological format and supports setting-up and using ontologies. Protégé uses a set of *basic terms*, which are implicitly built in. This basic terminology is similar, but not completely identical to the terminology used in the wider ontological literature and in other ontological tools. For the ontological framework discussed in this book a vocabulary will be described that provides ontological primitives to describe science and collaborative processes. To be convenient, the set of elements in this vocabulary should be small and sufficient for its purpose. To define this vocabulary normal English will be used. Definitions refer to each other or are axiomatic and understandable. Next to this vocabulary to describe the ontological framework, there are some additional *concepts*[1] that are needed. Typical examples of this category are the *paradigm describing aspects* related to monodisciplinary vocabulary (Kuhn's paradigm; Kuhn, 1970) and multidisciplinary problem solving (see Chapter 4). Vocabulary and concepts outside the scope of the three ontologies are called *meta-ontology*.

5.2 Existing bootstrapping vocabularies

Can we use some existing meta-ontology or should the meta-ontology be developed by a bootstrapping process, i.e. starting from scratch? OpenCyc[2] and SUO (including SUMO, i.e. Suggested Upper Merged Ontology)[3] are good examples of starting an ontology in a bootstrapping process, but these ontologies are very ambitious and beyond the scope of this book. Uschold *et al.* (1998) opted for a very practical approach by defining only the terms and concepts, essential for enterprise ontology in a simple meta-ontology. Another possible starting point can be found in Date's *semantic concepts*, used to bootstrap the relational database paradigm (Date, 2000). Other well-known examples of such bootstrapping terminology are those used for Ontolingua[4], which is based on KIF[5] and RDF[6]. Protégé's knowledge model is described in Noy *et al*. (2000). Its terminology is related to that of the ontological literature and is implicitly strongly linked to the terminology of the object-oriented paradigm in software engineering.

In Chapter 3 basic terms of ontologies are discussed: *ontology*, *concept*, *relation*, *property*, *function*, *axiom*, and *instance*. In order to determine whether any existing vocabulary can be used as a starting point of a meta-ontology, on top of which the three ontologies for multidisciplinary problem solving can be built, the following three sets of existing, bootstrapping terminologies will be compared: Date's *semantic concepts* for databases (Date, 2000), Uschold's *meta-ontology* for his enterprise ontology and the *glossary of ontology terminology*[7].

---

[1] *Concept* will be defined in the meta-ontology as one of the basic terms.
[2] http://www.opencyc.org/. The main objective of OpenCyc is to develop the world's largest and most complete general knowledge base and commonsense reasoning engine, based on Cyc, developed by the company Cycorp
[3] http://suo.ieee.org/index.html
[4] http://www.ksl.stanford.edu/software/ontolingua/.
[5] http://logic.stanford.edu/kif/kif.html
[6] http://www.w3.org/TR/rdf-concepts/
[7] http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/glossary-of-terms.html.

Table 5-1 The basic terms of Chapter 3 are compared with three bootstrapping terminologies, including the basic terms *ontology*, *concept*, *property*, *relation*, *sub-concept of*, *function*, *axiom*, and *instance*.

	Date, 2000	Uschold *et al*., 1998	Ontolingua[8]
Ontology	**Ontology:** Undefined. Comparable terms are: ER[9]-model (semantic data model) and ER-diagram (a technique for representing the logical structure of a database in a pictorial manner).	**Ontology:** Undefined.	**Ontology**: An ontology is an explicit specification of some topic. For our purposes, it is a formal and declarative representation, which includes the vocabulary (or names) for referring to the terms[10] in that subject area and the logical statements that describe what the terms are, how they are related to each other, and how they can or cannot be related to each other. Ontologies therefore provide a vocabulary for representing and communicating knowledge about some topic and a set of *relationships* that hold among the terms in that vocabulary. Note 1. A term is any object that has a definition (e.g., *slots*, *classes*, *instances*, *relations*, *functions*). An axiom is not considered a term.
Concept	**Entity type**: A distinguishable object. Note 1. *Entity* according to Date (2004) is an *object* in Ontolingua. Note 2. *Entity type* is the *class* of *entities* rather than just an *object*.	**Entity**: a fundamental thing in the domain being modeled. Note 1. Uschold *et al*., 1998 use *type* of ENTITY for entity and *particular* ENTITY for instance.	**Class**: A class is a representation for a conceptual grouping of similar terms.
Attribute	**Attribute** (also known as property): A piece of information that describes an *entity*.	**Attribute**: a *relationship* between two *entities* (referred to as the 'attributed' and 'value' *entities*) with the following property: within the scope of interest of the model, for any particular attributed *entity* the *relationship* may exist with only one value *entity*.	**Facet**: Facets are used to represent information about a *slot*[11] on an object.
Relation	**Relationship**: An *entity* that serves to interconnect two or more *entities*.	**Relationship**: the way that two or more *entities* can be associated with each other.	**Relation**: A *relation* is used to describe a relationship among two or more terms. If a relation represents a relationship between only two terms, it is called a *slot* or a *binary relation*. If the *relation* describes a relationship among *n* terms such that there is a unique nth term corresponding to any set of the first n-1 terms, then the *relation* is called a *function*.

[8] http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/glossary-of-terms.html.
[9] *ER* means *Entity Relation* and is used in combination with model (ER-model) and diagram (ER-diagram).
[10] *Term* is here used where in the majority of ontological literature *concept* is used.
[11] A *slot* is used to describe a *relationship* between two terms. The first term must be an *instance* of the *class* that is the 'domain' of the *slot* and the second must be an *instance* of the *class* that is the 'range' of the *slot*. For example, 'brother' could be represented as a *slot* such that its 'domain' was 'animal' and its 'range' was 'male-animal'. A *slot* may also be referred to as a 'binary *relation*'.

Sub-concept of	**Subtype of**: *Entity type* Y is a *subtype* of *entity type* X if and only if every Y is necessarily an X.	**Subclass of**: Undefined, but term used nevertheless, supposing an inheritance mechanism as in the Object Oriented paradigm.	**Subclass of**: Implicitly (but not explicitly) defined by 'Every *class* must be a *subclass of* some other *class*. Note 1: The slot super-class-of is the inverse[12] of subclass-of.
Function	**Functional Dependence**: A functional dependence (FD) is a many-to-one *relationship* from one set of *attributes* to another within a given *relation*. Note 1. More formally: Let R be a *relation*, and let X and Y be arbitrary subsets of R. Then we say that Y is functionally dependent on X, if and only if each X-value in R has associated with it precisely one Y-value in R.	**Attribute**: A *relationship* between two *entities* (referred to as the 'attributed' and 'value' *entities*) with the following property: within the scope of interest of the model, for any particular attributed entity the *relationship* may exist with only one value *entity*.	**Function**: A *function* is a special type of *relation*, which relates some number of terms to exactly one other term. That is, a *function* is a *relation* such that no two *relationships* of n terms in the relation have the same first n-1 terms.
Axiom	**Axiom**: Undefined, but used nevertheless.	**Axiom**: Undefined, but used nevertheless. They follow the Ontolingua definition, but without the Ontolingua restrictions, this because their emphasis was more on reducing ambiguity for humans than on automatic translation.	**Axiom**: An *axiom* is a sentence in first order logic that is assumed to be true without proof. In practice, we use *axioms* to refer to the sentences that cannot be represented using only *slots* and values on a frame.
Instance	**Instance**: Individual objects. Note 1. *Instance* is used to distinguish *objects* from *classes*, but *objects* and *instances* do not belong to the core relational model, but to its OO extension. In the core relational model Date uses *entity type* for *classes* (here called *concepts*) and *entity* for *objects* (here called *instance*).	**Instance**: a particular *entity*. Note 1. Uschold *et al*. use *type* of ENTITY for entity and *particular* ENTITY for instance.	**Instance**: All of the terms in an ontology that have an associated definition (i.e., *classes*, *slots*, *relations*, *functions*, facets) are an *instance* of some *class*. *Classes* are *instances* of *Class*, *functions* are *instances* of *Function*, etc. Note 1. *Facets* are used to represent information about a *slot* on an object. Usually *facets* represent some constraint on an instance slot. Note 2. See also *sub-class* and *super-class*.

Comparing basic terms in three bootstrapping terminologies shows that their meaning differs considerably (Table 5-1). Moreover, some terms are not defined or used to bootstrap the development of ontologies or databases. Not all bootstrapping terms from the bootstrapping terminologies are incorporated in Table 5-1 Most terms are related to some specific use, for which the terminologies are developed, being database development (Date, 2000), bootstrapping the Enterprise Ontology (Uschold *et al*., 1998) and general ontology development for information integration, knowledge-level interoperation, and knowledge-base development (Farquhar *et al*., 1996). Each of these terminologies opts for a focus on making the terminology particular for its intended purpose. This suggests that it is more efficient and effective to use a specific terminology for bootstrapping the meta-ontology to support the development of the three ontologies for multidisciplinary model-based problem solving. This approach has been followed in the remainder of this chapter, but my bootstrapping concepts, presented in section 5.4.1 are based on those presented in Table 5-1.

[12] The *inverse* of a *slot* is the *slot*, which relates the two terms in reverse order.

5.3 Design remarks on a bootstrapping meta-ontology

Before setting up a meta-ontology, design requirements will be discussed here. As stated this will not be done using some existing set of terms, but a new bootstrapping meta-ontology will be developed, based on the overview in Table 5-1, especially the meta-ontology of Uschold *et al*. (1998). The following design requirements have been used for this purpose:

1. The content of the meta-ontology should be as small as possible and is further determined by the following criteria:
   a. all terms from Table 5-1;
   b. other terms that are needed to describe ontologies in general, according to own experiences;
   c. all concepts needed to set-up (structure and knowledge content) the three other ontologies, i.e. a *modelling* ontology, an *object system and problem* ontology and a *model* ontology);
   d. all types of relations to be used for the meta-ontology and the three other ontologies in this book;
   e. the concepts linking multidisciplinary science with the other three ontologies;
2. The meta-ontology can best be structured to facilitate re-use of parts, with the following structure[13] (see Figure 5-1):
   - *Bootstrapping concepts*: vocabulary needed to describe the definitions in the meta-ontology (section 5.4.1);
   - *Bootstrapping relations*: relations needed for the set of three other ontologies (section 5.4.2);
   - *Basic meta-ontology*: concepts needed to describe the set of three other ontologies (section 5.4.3);
   - *Multidisciplinary meta-ontology*: concepts linking multidisciplinary science as described in Chapter 4 and the basic meta-ontology with the set of three other ontologies (section 5.4.4);
   - *Additional terminology*: concepts for decomposition and ontology development (section 5.4.5).

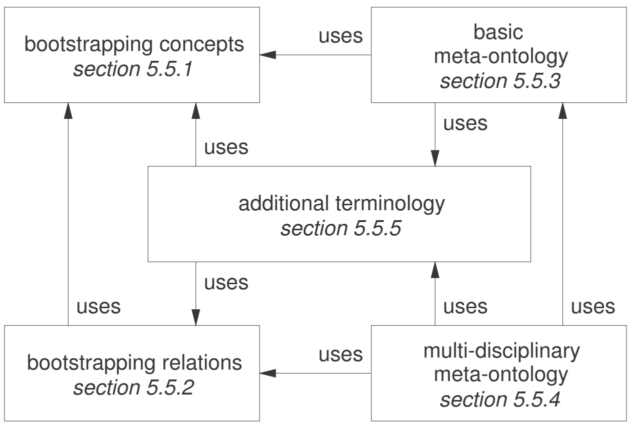


Figure 5-1. Components and structure of the meta-ontology; rectangles are *concepts* and arrows[14] are *relations*. The terms *concepts* and *relations* are defined in the meta-ontology. The only *relation* used here is *uses*[15].

The meta-ontology presented in section 5.4 is not actually implemented in Protégé, as this is not needed. Its content is only needed to bootstrap the other ontologies. An implementation will not be difficult and the only hurdles will be the following: (1) Protégé uses another terminology, which is more related to the terminology of Ontolingua (see Table 5-1) and (2) the bootstrapping meta-ontology terminology proposed here should be represented in a format significantly different from the one proposed in section 5.4.

In the subsequent section the meta-ontology will be presented, but not discussed. Many discussion issues on the meta-ontology can be found in the previous Chapter 4.

[13] The components, how they are related and their meaning can be classified as a meta-meta-ontology, as they describe what is used to set up the meta-ontology.
[14] Arrows have to be interpreted as sentences with the following grammar: {source of arrow = subject | arrow name = part of speech, usually an action, occurrence or state of being | destination of arrow =object}.
[15] *Uses* is defined in section 5.4.2, Table 5-3.

Ontologies have different types of purposes (Chapter 3), of which facilitating communication is a major one (Uschold and Grüninger, 1996). This communication can occur between *real persons* and *machines*[16][17]. Some parts of the communicated information should be understandable for machines (PC, computer program) and some parts should be understandable for human beings (knowledge base users, knowledge experts, knowledge engineers). The machine understandable parts of the ontologies in this book typically belong to the ontological structure and what real persons have to understand belongs to the ontological instances, i.e. the content of a knowledge base.

5.4 Meta-ontology

5.4.1 Bootstrapping concepts

The bootstrapping concepts, presented in Table 5-2 contain all terms from Table 5-1, plus some additional concepts, needed to define ontologies in general.

Table 5-2. Bootstrapping concepts. All terms in *italic* are defined components of the meta-ontology[18].

Concepts	Meaning
Ontology	1. An *ontology* is an explicit specification of a conceptualization (Gruber, 1993, 1995). 2. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. (Gruber, 1995). 3. A conceptualization refers to what can be represented in terms of *concepts* and the relationships among them, reflected in the representational vocabulary (Gruber, 1993). 4. An *ontology* is a formal specification of a shared conceptualization (Borst, 1997). 5. An *ontology* is a concise and precise, formal specification shared by its (human or machine) users and providing sufficient vocabulary such that a piece of knowledge can be formalized for its intended purpose (this book). 6. The main structure of an *ontology* is set-up by its *concepts* and the *relations* between the *concepts*. Note 1. 'Shared' includes the requirement that users of an ontology should have the same perception of some real thing (a concept and – broader – a ontology (Smith, 2004)).
Concept	*Concepts* can be abstract or concrete, elementary (electron) or composite (atom), real or fictitious. In short, a *concept* can be anything about which something is said, and, therefore, could also be the description of a task, *function*, *action*, strategy, reasoning *process*, etc. Note1. According to Corcho and Gomez-Perez (2000). Note 2. *Concepts* are also known as 'classes' (e.g. in Protégé, see also Table 5-1). Note 3. A *concept* is more universal than a *concept* / *instance* of a lower *ontological layer*[19] (Smith, 2004). Note 4. A *concept* includes Date's (2004) *entity type*.
Sub-*concept*	If a *concept* is a *sub-concept* of another *concept*, *inheritance* causes the *sub-concept* to inherit all characteristics of the *concept* of which it is a *sub-concept*. Note 1. See *inheritance*. Note 2. *Sub-concepts* are also known as 'subclasses' (e.g. in Protégé, see also Table 5-1).

[16] In fact, the following types of communication can occur: *from machine to machine*, *from machine to person*, *from person to machine* and *from person to person*. A machine is typically a computer or a computer program. If the receiver of to communicated information is a machine, the information should be formal and if the receiver is a person the information should be understandable for human beings.
[17] Both terms are defined in section 5.4.3, Table 5-1.
[18] Some are defined in the same part of the meta-ontology, some in other parts.
[19] *Ontological layer* is defined in section 5.4.5, Table 5-6.

Attribute	An 'element' of an *ontology* that describes other 'elements'. An *attribute* has at least a name and a value. Note 1. Adapted from Wikipedia (*element* replaced by *concept*): http://en.wikipedia.org/wiki/Ontology_%28computer_science%29. Note 2. In the terminology of this book 'element' is a synonym for *concept*.
Property	An *attribute* that is not a *relation*, but some characteristic of a *concept*.
Relation	1. An *attribute* that describes the *relation*ships (also known as *relations*) between *concepts* in the *ontology*. Typically a *relation* is an *attribute* whose value is another *concept* in the *ontology*. 2. *Relations* represent a type of interaction between *concepts* of the *domain*. They are formally defined as any subset of a product of *n* sets, that is: R: C_1 x C_2 x ...x C_n. 3. A *concept* that enables to define connections between two or more *concepts*. Note 1. Meaning 1 from http://en.wikipedia.org/wiki/Ontology_%28computer_science%29 (Wikipedia). Note 2. Meaning 2, according to Corcho and Gomez-Perez, 1999. Note 3. An example of a binary *relation* is: *part_of*. Note 4. *Relations* have a direction 'from_to', from concept1 to concept2, e.g. the *relation is_a* directs from *human_actor* to *actor*. Note 5. In this book *relations* can be divided in *hierarchical relations* and *property relations.*
Hierarchical *relation*	A *relation* connecting to concepts that are part of some hierarchy. Note 1. The term 'hierarchy' is not a part of this *meta-ontology*, but it can be defined (from Wikipedia: http://en.wikipedia.org/wiki/Hierarchy) as a *system of ranking and organizing things or people, where each element of the system (except for the top element) is subordinate to a single other element*. Examples: (from computer science) modularity of hardware systems, of hierarchical file-directories and of class hierarchies in object orientation, (biology) taxonomies, (from physics) the 'standard model' of fundamental particles and interactions.
Property *relation*	A *relation* connecting a *concept* with a *property*.
Precedence *relation*	A *relation* indicating the order of *concepts*. Note 1. In mathematics (and in programming languages) there is a standard order of operations: (1) exponents and roots, (2) multiplication and division and (3) addition and subtraction. Parentheses can change this order. Note 2. 'Order of concepts' means here the order in which a concept is relevant or should be executed.
Instance Instantiation	An *instance* is a specialization of a *concept* and *instantiation* is enacting an *instance* on a *concept*. Note 1. An *instance* is a *concept* that is not a 'type' of something; 'type' is here used according to Date (2004) and according to the Object Oriented paradigm. Note 2. An *instance* is more particular than a *concept* of the *ontological layer* above it (Smith, 2004); *instance* belongs to the lowest *ontological layer*. Note 3. An *instance* contains a piece of the knowledge.
Function	1. A many-to-one *relation*, resulting in one outcome. 2. Functions are a special case of relations in which the n^{th} element of the relationship is unique for the *n-1* preceding elements. Formally, functions are defined as: F: C_1 x C_2 x ... x $C_{n-1} \rightarrow C_n$. Note 1. Meaning 2 is a more formal definition from Corcho and Gomez-Perez, 1999.
Axiom	Piece of knowledge that cannot be proved. Typically a logical statement. Note 1. Axioms are not used in the ontologies in this book.

5.4.2 Bootstrapping relations

This section of the bootstrapping ontology consists of all relations used for the ontologies in this book. The rather abstract character of the relations allows this approach. The relations are not very specialized and do not contain much content related to the three ontologies (*modelling ontology*, *problem and object system* *ontology* and *model ontology*). In this way the relations defined here can be used in all other ontologies, as they are defined in the meta-ontology.

Table 5-3. Bootstrapping relations. All terms in *italic* are defined components of the meta-ontology.

Relations[20]	Meaning
Identical_to	*Hierarchical relation* denoting that a *concept* is identical to the related *concept*.
Is_a	*Hierarchical relation* denoting that a *concept* is a *concept* of a lower *ontological layer* of the related *concept* or it is an *instance of* the related concept.
Has	*Property relation* denoting that a *concept* has something or some aspect of another concept.
Part_of	*Hierarchical relation* denoting that a *concept* is a part of another *concept*.
Specialization_of	*Hierarchical relation* denoting that a *concept* is more particular and from a lower, more specialized and particular *ontological layer*.
Instance_of	*Hierarchical relation* denoting that a *concept* is a *specialization_of* another *concept* in a higher *ontological layer*.
Property_of	*Property relation* denoting one or more aspects of a *concept*.
Performed_by	*Property relation* denoting that a *concept* is performed by another *concept*, typically by an *actor*. Note 1. *Actor* is defined in the basic meta-ontology concepts of Table 5-4.
Determined_by	*Property relation* denoting that a *concept* is determined by another *concept*, in other words, the latter concept controls the structure and or content of the former.
Uses	*Property relation* denoting that a *concept* can use what is defined in another *concept*.
View_on	*Property relation* denoting a way of looking at situations or topics etc. Note 1. 'Situations and topics' refer to *concepts*.
Next	*Precedence relation* denoting the concept that follows after the present concept.
Previous	*Precedence relation* denoting the *concept* that precedes the present *concept*.
Feedback_to	*Precedence relation* denoting the *concept* that follows after the present *concept*; the *next concept* has earlier been the present *concept*. Note 1. This unusual definition is the result of negotiation to get agreement on it in the HarmoniQuA project.
Feedback_from	*Precedence relation* denoting the *concept* that precedes the present *concept*, but the present concept has been the present concept earlier. Note 1. This unusual definition is the result of negotiation to get agreement on it in the HarmoniQuA project.

5.4.3 Basic meta-ontology

The basic ontology *uses* the terminology provided by the bootstrapping concepts (Table 5-2) and the bootstrapping *relations* (Table 5-3) parts of the meta-*ontology*.

Table 5-4. Basic meta-ontology concepts. All terms in *italic* are defined components of the meta-ontology.

Some basic concepts	Meaning
Action	The state or process of acting or doing. Note 1. The American Heritage® Dictionary of the English Language, Fourth Edition. Note 2. The other meanings are inappropriate and are not used.
Single action	An *action* consisting of a single part, which is atomic in social and technical contexts and belongs to the *performance repertoire* of *actors* involved in the *action*.
Composite action	An *action* made up of complicated and related parts.

[20] Table 5-3 includes all *relations* used throughout the remainder of this book.

Actor	An entity that acts and gets things done; synonyms: doer, worker. Note 1. WordNet 2.1. Note 2. *Agent* is a synonym for *actor*.
Human *actor*	Human that executes some *action*.
Machine *actor*	Machine, typically a computer that executes some *action*.
Team	Group of *actors*, of which two or more are *human_actors*.
Process	1. (here) *Composite action* that is *performed_by* a *team*; 2. A sustained phenomenon or one marked by gradual changes through a series of states; 3. A sequence of steps performed for a given purpose; for example, the software development process. 4. A set of tasks to be performed, perhaps entirely by people or at least with some human involvement, and some form of ordering amongst them. Note 1. Process of modelling (including representing the relevant aspects in a model) should not be confused with processes that are modeled. This meaning is related to meaning 4. Note 2. Meaning 2 is common sense and typically used in science. Note 3. Meaning 3 comes from IEEE Std 610.12-1990 Note 4. Meaning 4 is by Dourish (2001).
Project	Planned *process*. Note 1. Project focuses on management and planning aspects (begin, end, resources, etc.), while process focuses on what has to be done.
Structure diagram	Diagram depicting the structural characteristics (of a part) of an *ontology*, i.e. the *concepts* and the *relations* between them.

Figure 5-2 depicts the structure of the basic meta-ontology of Table 5-4. This structure contains all that is needed to set-up the process ontology for modelling in Chapter 6. The structure, the definitions of the concepts and the definitions of the relations represent most elements of the basic meta-ontology.

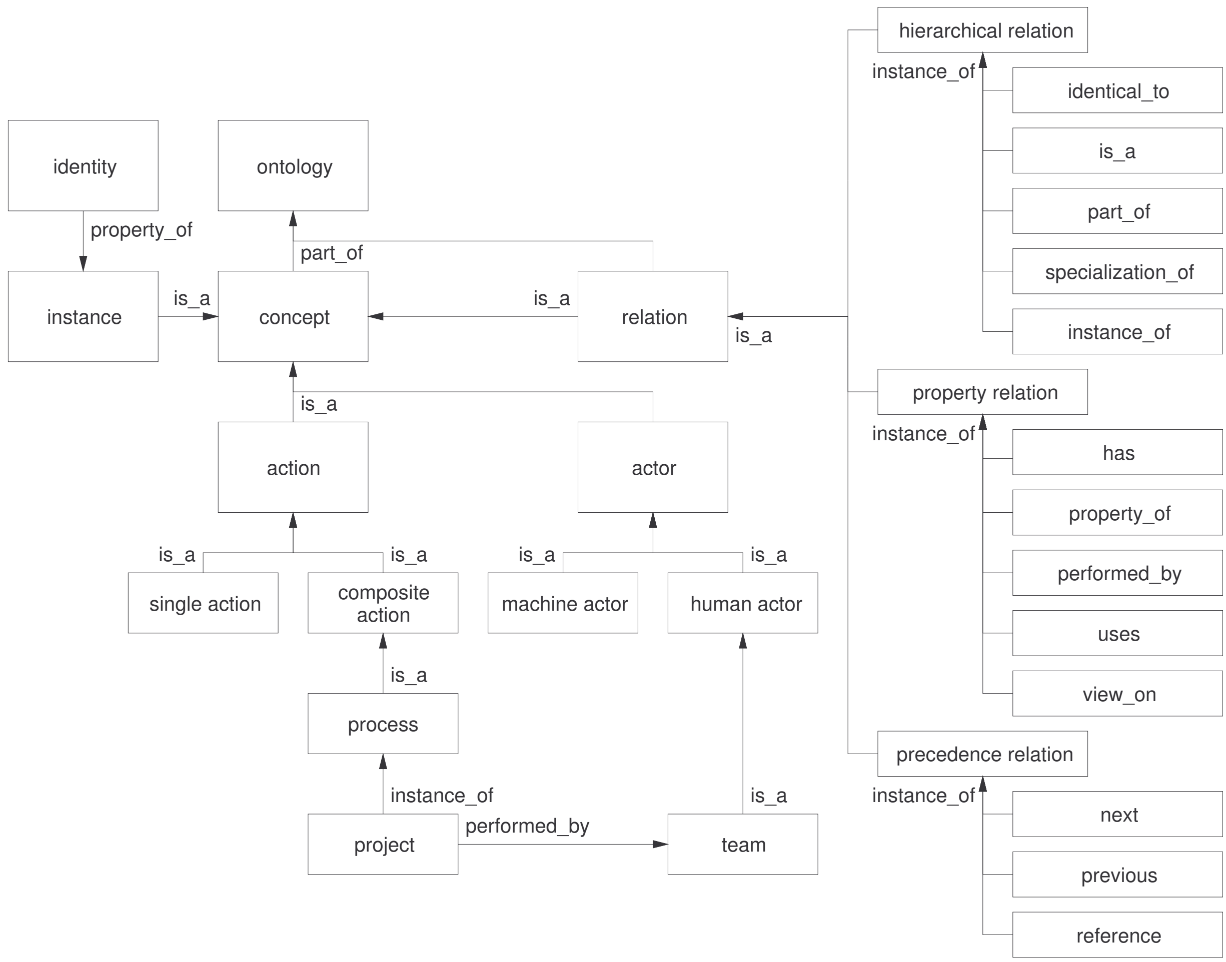


Figure 5-2. The structure of the basic meta-ontology structure, required for bootstrapping other ontologies in this book; rectangles are *concepts* and arrows are *relations*.

5.4.4 Multidisciplinary meta-ontology

Multidisciplinary science introduces an extra series of problems that were introduced and discussed in Chapter 4. This part of the meta-*ontology* describes the additional *concepts* and *relations* to describe multidisciplinary science.

Table 5-5. Multidisciplinary meta-ontology. All terms in *italic* are defined components of the meta-ontology.

Concepts	Meaning (concepts in *italic*)
Science	1. A particular branch of scientific knowledge; 2. Ability to produce solutions in some problem *domain*; Note 1. From the Webster Online Dictionary.
Scientist	*Human actor* involved in *science*.
Discipline	Field of study, a branch of knowledge, which is taught or researched at college or university level. Note 1. From Wikipedia: http://en.wikipedia.org/wiki/Academic_discipline. Note 2. *Discipline, domain* and *paradigm* are often used as synonyms; here *discipline* highlights the educational background of the researcher; *domain* stresses the field of application; *paradigm* emphasizes Kuhn's use of the *concept* (Kuhn, 1970), which is discussed in Chapter 4.
Disciplinary	Relating to a *discipline*.
Multi*disciplinary*	Relating to more than one *discipline*.
Mono*disciplinary science*	Science related to a single *discipline*.

Multi*disciplinary science*	*Science* related to more than one *discipline*.
Multi*disciplinary team*	A *team* of *human actors* originating from different *paradigm*s, typically with different *roles* in a model-based problem solving project. Note 1. This term is one of the *paradigm describing aspects*; see Chapter 4.
Paradigm	1. What is to be observed and scrutinized. 2. The kind of questions that are supposed to be asked and probed for answers in relation to this subject. 3. How these questions are to be put. 4. How the results of scientific investigations should be interpreted. 5. What a *scientific community* shares. It consists of: • a shared disciplinary background; • a shared *world view*; • shared *perceptions, concepts and terminology*; • agreement on which *theories and hypotheses* are accepted; • shared *general assumptions*; • agreement on accepted *methodology*; • a shared set of *exemplars*. Note 1. From Thomas Kuhn in: The Structure of Scientific Revolutions (Kuhn, 1970). Note 2. *Discipline, domain* and *paradigm* are often used as synonyms; here *discipline* highlights the educational background of the researcher; *domain* stresses the field of application; *paradigm* emphasizes Kuhn's use of the *concept* (Kuhn, 1970), which is discussed in Chapter 4. Note 3. Throughout this book the 5th meaning will be used; see also Chapter 4.
Domain	(Scientific) field of application. Note 1. *Discipline, domain* and *paradigm* are often used as synonyms; here *discipline* highlights the educational background of the researcher; *domain* stresses the field of application; *paradigm* emphasizes Kuhn's use of the *concept* (Kuhn, 1970), which is discussed in Chapter 4. Note 2. Often referred to as *domain of application*. Note 2. Examples: water management, environmental science, supply chain management.
Sub*domain*	Part of a *domain*; Note 1. Example: hydrodynamics is a sub*domain* of the *domain* water management.
Paradigm describing aspects	A set of characteristics describing a *paradigm*
Scientific community	*Paradigm describing aspect* referring to a group of *scientists* within a single *paradigm*. Note 1. This term is one of the paradigm describing aspects; see Chapter 4. Note 2. Sometimes 'professional community' is used as synonym.
Common worldview	*Paradigm describing aspect* consisting of shared non-scientific concepts, including non-scientific socio-economic aspects and general scientific concepts, which not strictly belong to the paradigm. Note 1. This term is one of the paradigm describing aspects; see Chapter 4.
Perceptions, concepts, terminology	*Paradigm describing aspect* consisting of perceptions, concepts, terminology, belonging to the scientific paradigm and shared by scientists within the paradigm. Note 1. This term is one of the paradigm describing aspects; see Chapter 4.
Theories and hypotheses	*Paradigm describing aspect* consisting of a set of consistent theories, thoroughly tested and published, accompanied by less tested hypotheses, which did not reach the status of theories yet. Note 1. This term is one of the *paradigm describing aspects*; see Chapter 4.

General assumptions	*Paradigm describing aspect* consisting of a set of generic assumptions, i.e. not accounting for non-scientific social aspects, which are included in *worldview*, but assumptions required for the consistency of the *theories and hypotheses* and *methods* and therefore boundary conditions for a *paradigm*. Note 1. This term is one of the *paradigm describing aspects*; see Chapter 4.
Practices	*Paradigm describing aspect* consisting of the set of practices needed to apply *methods* to solve problems that fit directly into the *exemplar* straitjacket or that can be solved within the paradigm after some adaptations. Note 1. This term is one of the *paradigm describing aspects*; see Chapter 4.
Accepted *practices*	*Practices* that are accepted as sufficiently good by the professional community (not necessarily 'best practices'.
Performance repertoire	Possible *actions* that refer to what *actors* can do and are used to do.
Methodology	*Paradigm describing aspect* consisting of the set of accepted *methods* are the result of a *process* consisting of proposing and testing methods according to the shared scientific practice, often applied on problems that are similar to *exemplars* or that can be molded in the form *exemplars* have; these methods are discussed on conferences visited by the group and published in journals/textbooks by the group; together with the accepted theories it forms the heart of a paradigm. Note 1. This term is one of the *paradigm describing aspects*; see Chapter 4.
Exemplars	*Paradigm describing aspect* consisting of rather simple problems, which fit perfectly in the paradigm, which can be solved with the methods of the paradigm and which solutions do not contradict with theoretical foundations of the paradigm; exemplars can best be seen as textbook problems and certainly not as real world problems. Note 1. Real world problems are often multidisciplinary and should not be confused with exemplars. Note 2. This term is one of the *paradigm describing aspects*; see Chapter 4.
Assumptions for *exemplars*	*Paradigm describing aspect* consisting of a set of (detailed) assumptions that transform and simplify a real problem into an exemplar. Notes 1. Assumptions for previously solved real problems will usually not be adequate for new real problems; lists of assumptions of comparable, previously solved real problems may act as starting point for a set of assumptions for new real problems. Note 2. This term is one of the paradigm describing aspects; see Chapter 4.
Model	To an observer B, an object A^* is a model of an object A to the extent that B can use A^* to answer questions that interest him about A. Note 1. From Minsky (1965). Note 2. See also *first draft of a mental process model*, given in Chapter 2, Table 2-1.
*Model*ling	A *process* focused on developing, specifying, analyzing and/or using a *model*.
Problem	An obstacle which makes it difficult to achieve a desired goal, objective or purpose. It refers to a situation, condition, or issue that is as yet unresolved. In a broad sense, a *problem* exists when an *actor* becomes aware of a significant difference between what actually is and what is desired. Every problem asks for an answer or solution. Note 1. From Wikipedia: http://en.wikipedia.org/wiki/Problem.
Problem solving	Finding a solution for or understanding a *problem*.
Problem owner	*Actor* who has a *problem*, typically a member of a *multidisciplinary team*.

Real-world *problem*	A *problem* which is not a research question and not an *exemplar*, but refers to a complex *problem*, as can be found the context of the real world. Note 1. A *research question* acts as the guiding force behind an experiment. It is the broad question that the experiment is supposed to answer. The research question poses the problem of the relationship between the objective(s) and the purpose, between the specific experimental procedure and why one is carrying out that procedure in the first place.
System	1. $\boldsymbol{S} = (\boldsymbol{T},\boldsymbol{R})$ with: $\boldsymbol{S}$ a system $\boldsymbol{T}$ a set of certain things (thinghood) $\boldsymbol{R}$ a relation defined on T (systemhood) 2. A group, set or aggregate of things, natural or artificial, forming a connected or complex whole. Note 1. Definition 1 is from Klir, 1991. Note 2. Definition 2 is from the Oxford English Dictionary. Note 3. Combining definitions 1. and 2.: A *system* consists of entities/things which are related to each other or which interact to each other. The *system boundary* divides *entities* (things) that are part of the *system*, and *entities* (things) that are outside of the *system* and therefore part of the environment of the system Note 4. *System boundary* is assumed to be equivalent with *object system boundary*, which is defined in chapter 8.
Object *system*	The *entities* (things) and *processes* in the real world or some defined *system* that will be the object of *modelling*.
Role	*Role* that a member of a *multidisciplinary team* in a *model*-based *problem solving* project plays to solve *problem*s using *model*s. Note 1. *Instances* of *roles*, relevant for multidisciplinary model-based problem solving are: *modeler*, *manager / client*, *auditor*, *stakeholder*, *public*. For other processes other roles are relevant. Note 2. *Instances* of *roles* are discussed in Chapter 6.

Figure 5-3 depicts the concepts of Table 5-5 and the relations between these concepts. Some (detailed) concepts are left out here, but can be find in Figures 4-1 and 4-2.

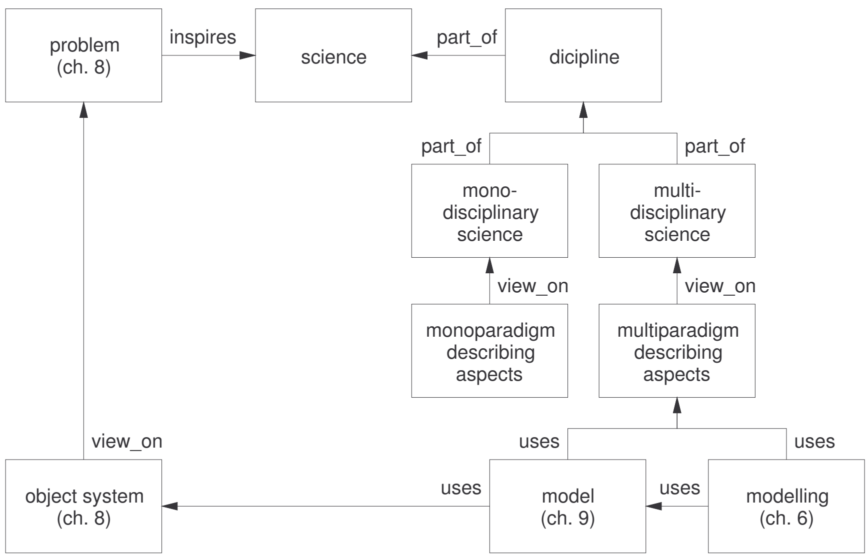


Figure 5-3. The multidisciplinary meta-ontology; rectangles are *concepts* and arrows are *relations*. The *concepts monoparadigm describing aspects* and *multiparadigm describing aspects* are discussed in detail in Chapter 4 and many details of these *concepts* are left out here.

5.4.5 Additional terminology

Because the proposed approach (methodology, ontologies and tools) aim at efficient and effective support of multidisciplinary teams in projects of various natures, an ontological framework will be designed with *ontological layers* of increased specialization, starting from a very abstract one with little content to a very specialized one carrying much content. Key issues here are *abstraction*, *specialization* and *classification*. These terms will be defined in section 5.4.5, Table 5-6.

Table 5-6. Additional terminology of the meta-ontology. All terms in italic are defined components of the meta-ontology.

Some basic concepts	Meaning
Inheritance	A way to take over (or inherit) characteristics, i.e. *attributes* or *properties* through a *property relation*, from an existing *concept*. It is intended to help reuse existing ontological *concepts* with little or no modification. Note 1. *Inheritance* can be compared with inheritance between classes in object-oriented programming.
Identity	The set of properties by which an object is recognizable as a member of a concept. Note 1. The concept *identity* makes a *specific* table out of 'a table', e.g. 'Riet's table'. Note 2. Adapted from http://dictionary.com.
Specialization Specialized	1. A concept with more semantic content and more properties. 2. Making a concept more special, i.e. with more semantic content and more properties. Note 1. An *instance* is a *specialization* of a *concept*. Note 2. A *specialization* is of a lower *ontological layer* than the *concept*, of which it is a *specialization*.
Generalization General	1. The inverse of a specialization, meaning 1, i.e. a concept with less semantic content and less properties. 2. A fact about the whole (as opposed to specific). Note 1. 'Inverse' is used in the mathematical sense, as in function *a* is the inverse of function *b*. Note 2. Meaning 1 is a definition in the context of this book.
Abstraction Abstract	1. A concept or idea not associated with any specific instance. 2. The process of formulating general *concepts* by abstracting common properties of *instances* (synonym: *generalization*). 3. A general concept formed by extracting common features from specific examples. Note 1. All meanings from WordNet 2.1. Note 2. The opposite of instantiation. Note 2. Example in meaning 3 refers to an object, i.e. concept or class.
Aggregation Aggregate	1. Several items grouped together or considered as a whole. 2. The act of gathering something together. Note 1. All meanings from WordNet 2.1.
Classification	The act of distributing items into classes or categories of the same type. Note 1. For Classes see Table 5-1, column Ontolingua. Note 2. Synonym: categorization. Note 3. Developing ontologies is a sort of classification.

Ontological layer	An *ontology* is supposed to be structured in layers; *concepts* in a higher *ontological layer* are more general than *concepts* in a lower *ontological layer*; lower *ontological layer concepts* are *specializations* and are often called *instances*. In this way *specialization* hierarchies can be defined with in the highest layer abstractions and in the lowest layer specializations. Note 1. 'Moby Dick' *is_a* specific 'whale'; a 'whale' *is_a* 'mammal'; a 'mammal' *is_a* 'vertebrate'. 'Moby Dick' *is_a specialization* of the *concept* 'whale', 'whale' *is_a specialization* of the *concept* 'mammal' and 'mammal' *is_a specialization* of the *concept* 'vertebrate'. Note 2. See also Figure 4-6.

Figure 5-4 depicts the concepts of Table 5-6 and the relations between these concepts. All concepts are linked to decomposition and how to represent hierarchical networks in an ontological setting.

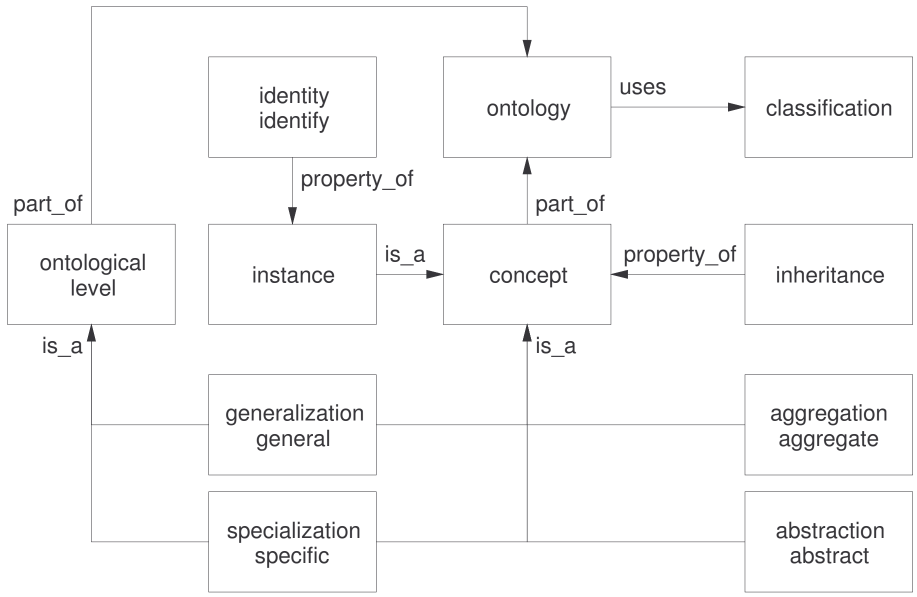


Figure 5-4. Additional terminology of the meta-ontology; rectangles are *concepts* and arrows are *relations*.

5.5 Conclusion

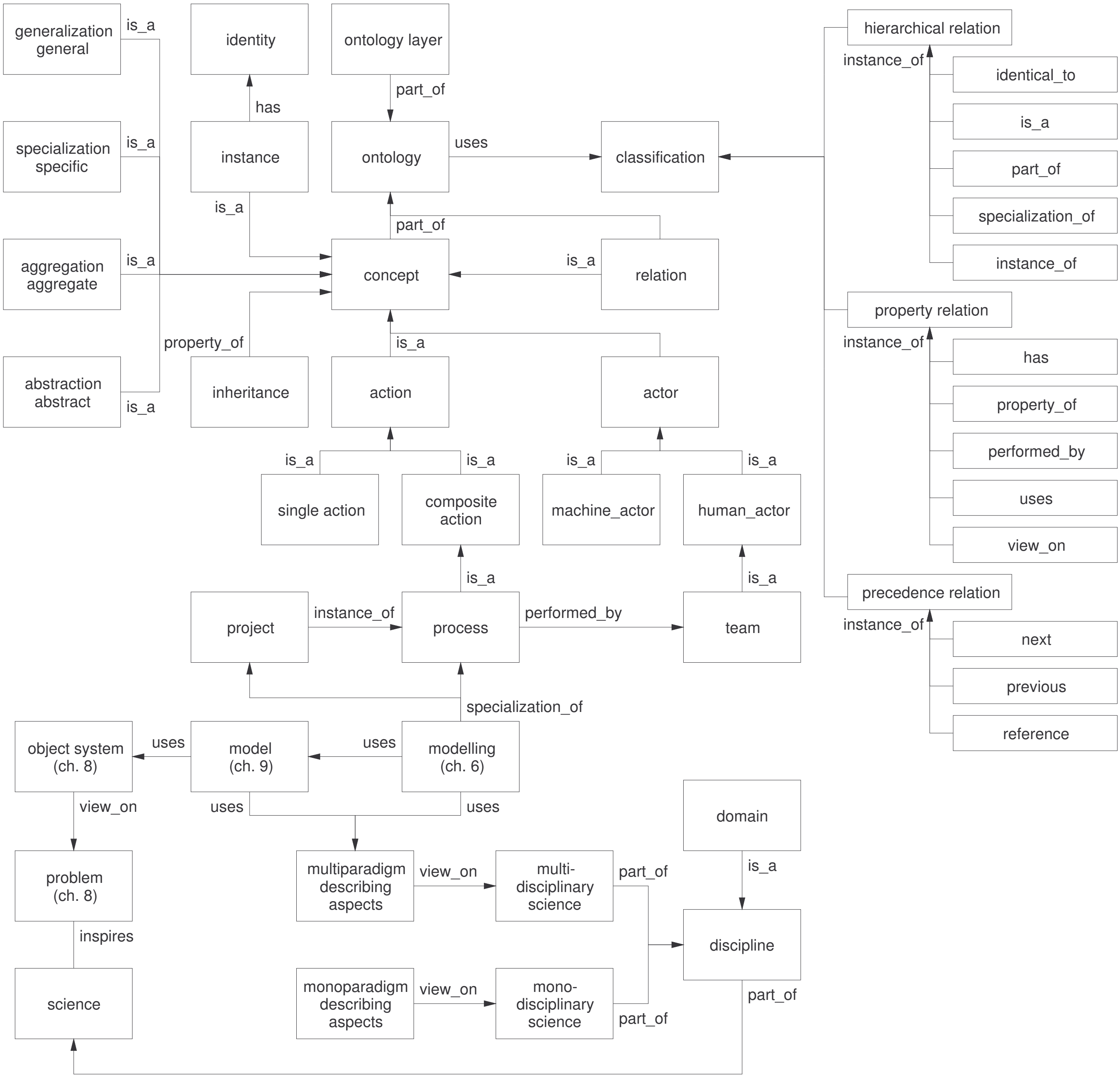


Figure 5-5. Overview of the meta-ontology; rectangles are concepts and arrows are relations. The concepts monoparadigm describing aspects and multiparadigm describing aspects are discussed in detail in Chapter 4 and many details of these concepts are left out here.

The meta-ontological framework presented in this chapter, of which Figure 5-5 shows an overview, does not pretend to be an ‘upper-ontology’ providing generic, common sense knowledge for a wide range of (domain) ontologies and their associated tools/applications. It merely enables bootstrapping the development of ontologies to support multidisciplinary model-based problem solving. The next 4 chapters will extend the presented ontological framework with many details, as is depicted in Figure 5-6. In Chapter 6 an ontology for modelling will be proposed, in Chapter 8 an ontology for problem description and its associated object system will be presented and Chapter 9 will focus on an ontology for (simulation) models. Chapter 7 will discuss tools for model-based water management using the modelling ontology. The presented framework in this chapter is a minimal meta-ontology, providing a knowledge context for the other ontologies discussed in the next four chapters.

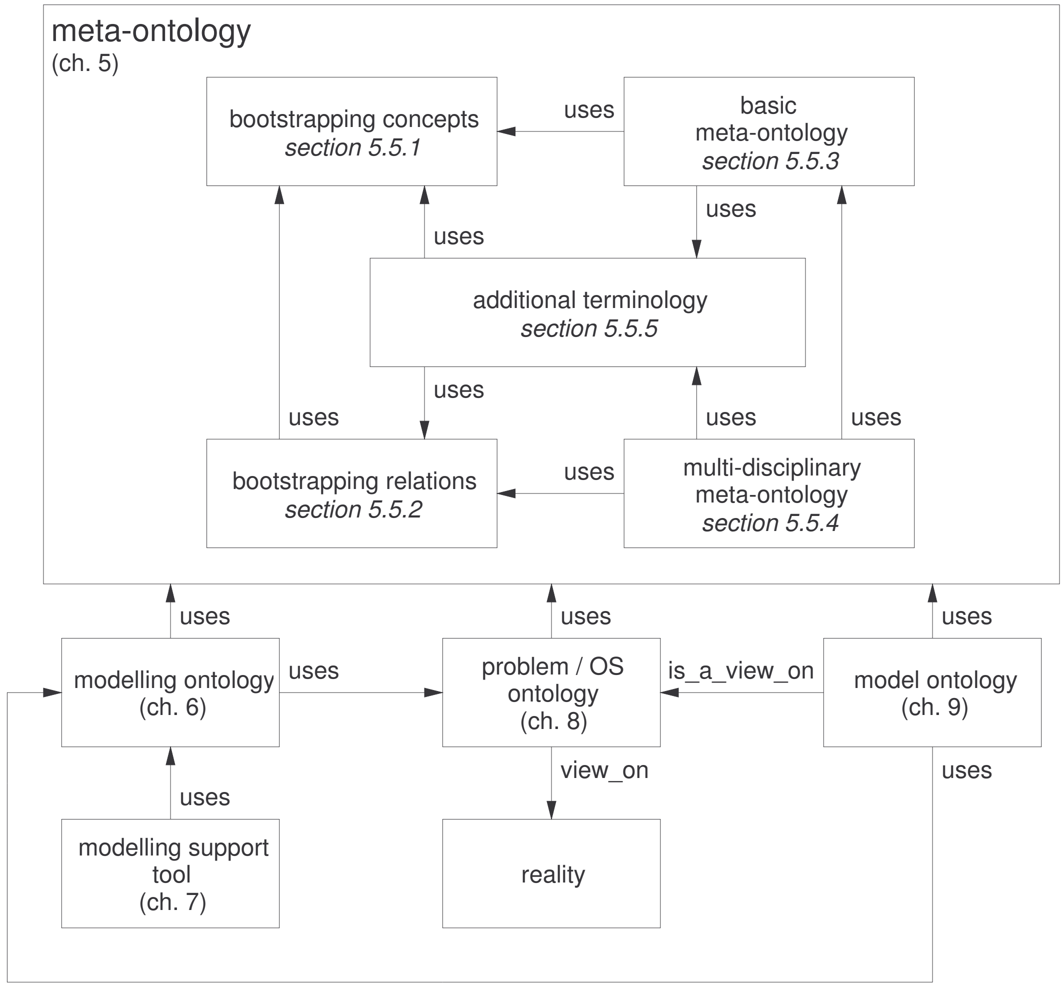


Figure 5-6. Links between the ontologies in the ontological framework of this book; rectangles are *concepts* and arrows are *relations*.

5.6 References

Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. University of Twente, The Netherlands, Enschede.

Corcho, O. and A. Gomez-Perez, 1999, Guidelines to Study Differences in Expressiveness between Ontology Specification Languages: A Case Of Study, Knowledge Acquisition Workshop, Banff, October 16-22, 1999.

Corcho, O., and A. Gomez-Perez. 2000. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. in ECAI-00 Workshop on Applications of Ontologies and Problem solving Methods, 2000, Berlin.

Date, C.J., 2004. An introduction to database systems. Addison-Wesley Publishing Company, Reading etc., 8th Edition, ISBN 0321197844.

Dourish, P. 2001. Process descriptions as organisational accounting devices: the dual use of workflow technologies. Pages 52-60 in Proceedings of the 2001 International ACM SIGROUP Conference on Supporting Group Work, October 2001. ACM, Boulder, Colorado.

Farquhar, A., R. Fikes, and J. Rice. 1996. The Ontolingua server: A tool for collaborative ontology construction. Report, Number, Knowledge Systems Laboratory, Stanford.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition 5((2)): 199-220.

Gruber, T. R. (1995). Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43:907-928.

Klir, G.J., 1991. Facets of system science, Plenum Publ., New York, 664 pp.

Kuhn, T., 1970. The structure of scientific revolutions. University of Chicago Press, Chicago, 210 pp.

Minsky, M. L. 1965. Matter, mind, and models. Pages 45-49 in Proceedings of IFIP Congress. Spartan Books.

Noy, N.F., R.W. Fergerson and M.A. Musen, 2000, The knowledge model of Protege-2000: Combining interoperability and flexibility, 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-2000-0830.pdf.

Smith, B. 2004. Beyond concepts: ontology as reality representation. In A. Varzi and L. Vieu, editors. Proceedings of FOIS 2004. International Conference on Formal Ontology and Information Systems, Turin.

Uschold, M. and M. Grüninger, 1996. Ontologies: principles, methods and applications. The Knowledge Engineering Review 11, 93-136.

Uschold, M., M. King, S. Moralee, and Y. Zorgios. 1998. The Enterprise Ontology.

6 A process ontology for multidisciplinary model-based problem solving instantiated for water management

The important thing in science is not so much to obtain new facts
as to discover new ways of thinking about them.
(Sir William Henry Bragg, 1862-1942)

Content of Chapter 6

This chapter is based on parts of the following papers:

Overall idea and implementation	Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.
Architecture	Scholten, H., A. Kassahun and J.C. Refsgaard, 2006, Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006, Research Publishing, Volume 3, 2231-2238.
KB-editor	Kassahun, A. and H. Scholten, 2006, A knowledge base system for multidisciplinary model-based water management, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006, ISBN 1-4243-0852-6 (978-1-4243-0852-1), http://www.iemss.org/iemss2006/sessions/all.html.
Use and reuse of technology	Scholten, H. and A. Kassahun, 2006, Supporting multidisciplinary model-based water management projects: a user perspective, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006, ISBN 1-4243-0852-6 (978-1-4243-0852-1), http://www.iemss.org/iemss2006/sessions/all.html.

6.1 Introduction

In Chapter 4 a structure has been proposed to organize knowledge for modelling in accordance with a two dimensional conceptualization. The first dimension refers to the three ontologies, which aim to structure knowledge categories for model-based problem solving: a *meta-ontology* providing vocabulary for the other ontologies, a *problem and object system ontology* (defining the problem at hand and an abstraction of the relevant aspects and their interactions), a *model ontology* (formalizing relevant aspects in a mathematical format) and a *modelling ontology* (defining what to do and how to do it). The second dimension refers to *ontological layers*, in which each of the ontologies can be organized. This ontological framework is summarized in Figure 6-1. This layered structure facilitates reuse of the more general parts by replacing parts of the ontological framework or by extending it. Reuse will be discussed in this chapter, Chapter 8 and Chapter 9.

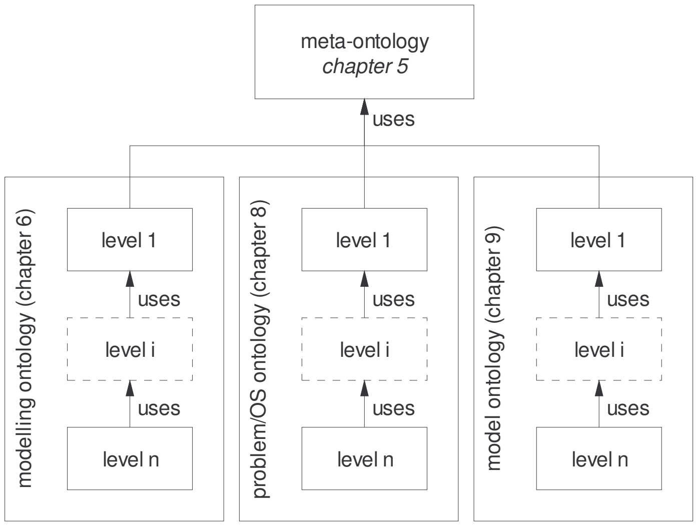


Figure 6-1. The ontological framework with the two dimensions (4 ontologies, each with *n* ontological layers of increasing specialization, of which 3 layers are depicted, i.e. *layer 1*, *layer i* and *layer n*. See text for more explanation.

The terminology to bootstrap the development of the three main ontologies (modelling, problem/OS and model) is proposed in Chapter 5. Here, some of the most important concepts will be recapitulated. Modelling is a *process*[1] executed by a *team* consisting of *actors* to solve some *problem* with (mathematical) *models*. Members of such *multidisciplinary teams* for *problem solving* have different *disciplinary* backgrounds and play different *roles*[2].

The main objective of this chapter is to develop an ontology for the modelling process itself. The ontology and its associated knowledge base about modelling will subsequently be used as part of a comprehensive support tool for teams involved in multidisciplinary model-based problem solving. This tool will be discussed in Chapter 7.

This Chapter 6 will discuss how to get to an ontological structure for model-based problem solving like simulation. In Chapter 2 introductions were given in related technologies, e.g. workflow management, process definition methods, ISO-standards, modelling guidelines and project management. The next section will discuss which technology is needed for defining processes in the modelling ontology. Next, the mental process model on model-based problem solving, introduced in Chapter 2, Table 2-1, will be updated[3] twice (section 6.3 and section 6.5). Each version of this mental model will give more details of my view on processes in general and modelling specifically. It also sets out the requirements for ontology based process support as presented in Chapter 6 and Chapter 7. Subsequently, design aspects of an ontological modelling

[1] A set of tasks to be performed, perhaps entirely by people or at least with some human involvement, and some form of ordering amongst them (Dourish, 2001). In this chapter and in the bootstrap ontology of chapter 5 the following, more precise definition, is used: *composite action* that is *performed_by* a *team*; it can be a chain of things to do, but typically it is a network of things to do, here of *modelling* tasks.
[2] Even if a *multidisciplinary team* in *modelling* consist of a single *actor*, this team member will usually have more *roles*, e.g. *problem owner*, *modeller*, *auditor*, and has to work in more than one (*disciplinary*) *domain*.
[3] In chapter 7 this mental process model for model-based problem solving will be updated a final time.

knowledge base (KB) will be discussed (section 6.4) followed by the structure of the ontological KB (section 6.6). The latter section starts with the terminology and bootstrapping framework of Chapter 5 and builds ontological layers of increasing specialization on top of this: *ontological layer 1* with *generic process knowledge*, *ontological layer 2* with *modelling knowledge*, *ontological layer 3* with *simulation modelling knowledge*, *ontological layer 4* with *simulation modelling ontology for (multidisciplinary model-based) water management.* The *ontological layer 5* contains the ontological structure for journals that log what actually has been done in projects (e.g. in instantiated processes like modelling) and instances of such journals. A summarizing outline of this construction is presented in Figure 6-2.

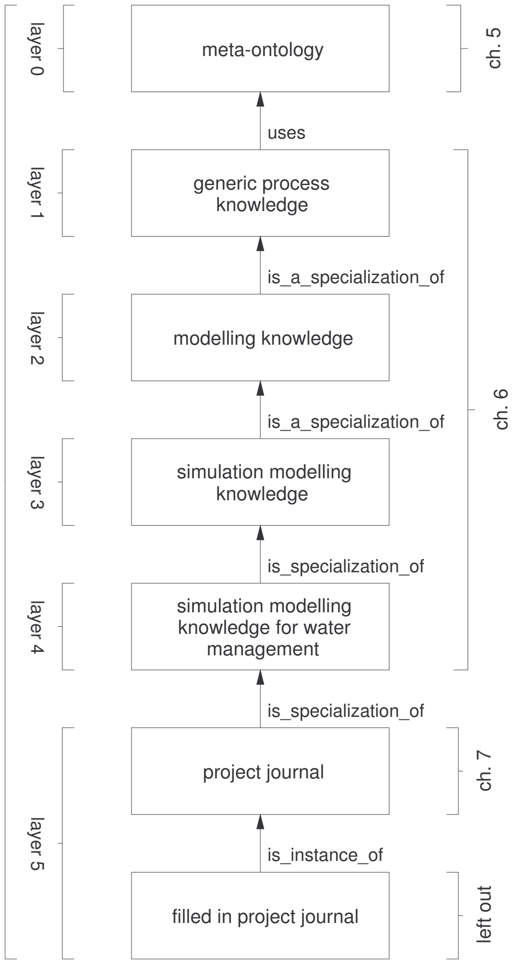


Figure 6-2. A stepwise ontology *specialization* with the modelling ontology expanded to some more specialized ontological *concepts*. The top (meta-ontology) is the most *generic ontological layer* and the *concepts* at the bottom the most *specific* ones. An *instance* is more *specialized* than the *concept* where it is based on. This *structure diagram* is a simplification of Figure 4-6.

6.2 Which technology to define processes?

Chapter 2 outlines major initiatives in formalizing and defining processes. So why not selecting the best (i.e. most appropriate) and use it to develop a framework of three ontologies (modelling, problem/OS, model) for multidisciplinary model-based problem solving? In this section the motivation to design a novel ontological technology for process definitions will be discussed.

Most process definition languages (e.g. PSL, Petri Nets, PDDL, PIF and UML, see Chapter 2, section 2.8.2) are not ontology based, which hinders flexibility, i.e. changing a process definition is more difficult. Furthermore, developing a proprietary, ontology based, technology (method + tools) will be more effective and more efficient, as resulting ontologies will be as uncomplicated as possible. Finally, choosing for a new, ontological, approach allows the designing of an ontological framework which includes ontological specialization layers (see meta-ontology in Chapter 5), which facilitate re-use of (higher layers of) the ontological framework for other processes, as will be explained in later chapters and is depicted in Figure 4-6.

Workflow management software comes closest to what is needed to support (modelling) processes with its process definition phase and its project execution phase. But workflow management systems are typically designed for standardized processes executed frequently and often many of them in parallel at the same time. Complex processes like modelling have different requirements, e.g. including more guidance and more support for co-operation. Using a workflow management platform is a possible approach but this would hinder a tailored handling of functional requirements particular to the design of modelling support software.

The development of the Good Modelling Practice Handbook (Van Waveren *et al*., 1999, Scholten *et al*., 2000, Scholten, 2001, Scholten *et al*., 2001) and similar approaches were promising, but they all lacked flexibility and ease of use. Guidance provided by handbooks is adequately customized to modelling, but adding more support than just book-formatted guidelines asks for a completely different approach, as is chosen in this publication.

Project management software includes some of the functionality also found in workflow management software and some of it is also needed for modelling support, i.e. decomposing and describing the process ('project' in project management terms). Its functionality is focused on management issues (what is the status of all project components, what are the financial and time resource risks and how to deal with them), but project management software does hardly provide support of the non-management work in projects.

As none of the technologies presented in Chapter 2 are appropriate candidates to be used for multidisciplinary modelling support in the sense of this book, this chapter will outline a new technology, which enables one to define processes in general and modelling processes in particular. The development of a process ontology will be discussed, focusing on modelling as a process, organized in a modelling knowledge base. The knowledge base (KB) as such is of value as far as its content is the result of discussion and consensus of professionals in the field. It gains in value, if it can be used during modelling projects by multidisciplinary teams. A set of focused tools can help by adding the necessary functionality to set-up and fill the KB and provide the rest of the functionality to guide the modelling and keep records of what is done in modelling projects. The requirements for such a system, outlined in Figure 6-3, will be discussed in the remainder of Chapter 6 (ontological Modelling Knowledge Base and Knowledge Editor) and Chapter 7 (Modelling Support Tool, ontological Modelling Journal). The Knowledge Base and tools are intended to be used for many modelling projects in water management and – after extension – for other types of model-based problem solving projects.

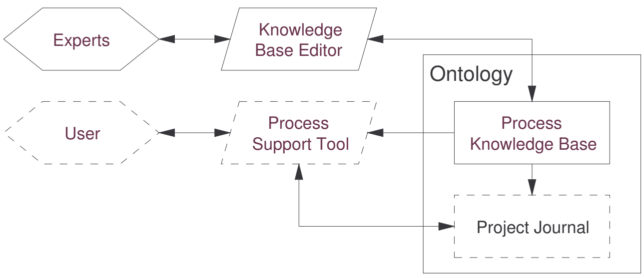


Figure 6-3. Outline of a software platform and KB to support modelling. Dotted lined shapes will be discussed in Chapter 7.

6.3 First draft of a mental model of model-based problem solving

Before discussing how to design a system as outlined in Figure 6-3, a *mental model of model-based problem solving* will be presented representing how professionals think about multidisciplinary problem solving in a project with a team of experts. This mental model is summarized in Table 6-1 and will be updated in section 6.5. In chapter7 the mental model will be extended to be fit for a tool to support the full multidisciplinary model-based water management cycle. This final, most detailed version, specialized for model-based water management is – to a large extent – also applicable to problem solving in general or even more generic processes of a collaborative type, as will be discussed in section 6.6.3 and in Chapters 10 and 11.

Table 6-1 First draft of the 'mental process model' summarizing how professionals view at multidisciplinary model-based problem solving[4]. Terms in *italic* are defined in the meta-ontology (Chapter 5).

Mental model of model-based problem solving, version 1

1. There is a *problem* to be solved, owned by some *problem owner*.
2. The *problem owner* organizes a *multidisciplinary team*, consisting of *team* members, who:
    a. Have different *disciplinary* backgrounds and
    b. Fulfill different *roles* in that *team*.
3. What *multidisciplinary teams* have to do ('solving' the problem at hand) is called a *process*, and is carried out in a *project* (in this chapter specifically a *modelling project*, instantiated for water management);
4. A *process* (a *composite action*) should be decomposed[5] into *single actions* that fit in the *performance repertoire* of *actors* involved in the *action*. A *process* should match *accepted practices* of professionals in the field.
5. *Practices* used in a *process* represent a body of (explicit and tacit) knowledge, often made partly explicit in textbooks and other publications, but typically not shared between the disciplines involved in solving the problem;
6. *Processes* are often *multidisciplinary*, i.e. cover more than one *domain*.

6.4 Towards an ontological process knowledge base

6.4.1 System outline

The design of process support in general and of modelling support specifically is focused on users and their demands and how they can be served and supported throughout a (modelling) project. The discussed system is strongly team oriented. Nowadays many model-based problem solving projects are of a multidisciplinary nature. Team members with different (disciplinary) backgrounds, collaborate in projects and this introduces new problems associated with the interaction between team members. In addition to the complexity caused by a different background of the team members, they also have to play also different roles in a project. To solve or control this type of problems a system has to be designed based on several particular components, which will be outlined here briefly.

The modelling support system proposed here should at least guide the modelling team and monitor what they do, which resembles to some extent the functionality of workflow management system. The role of ontologies is twofold: it provides a structure for the guidance part (i.e. the modelling knowledge) and it provides a more or less similar structure for the work done and monitored by the system and filed in a modelling archive[6] consisting of modelling journals[7]. A sketch of the modelling KB and modelling support tool is given in Figure 6-4. The two most important functionalities of such a system, guidance and monitoring, have already been proposed earlier (Scholten, 1994, Scholten and Udink ten Cate, 1995, 1996, 1999), but these earlier designs were simpler and did not account for a multidisciplinary nature of modelling projects and lacked the flexibility of an ontological approach. The guidance part has already been implemented for a series of guidelines. Examples of modelling guidelines for water management are discussed in Refsgaard and Henriksen (2004) and Refsgaard *et al.* (2005). In the Netherlands the GMP Handbook and the norms for modelling in water management (Chapter 2, section 2.5) are examples of such guidelines (Van Waveren *et al*., 1999, Scholten *et al*., 2000, Scholten, 2001, Scholten *et al*., 2001).

---

[4] The presented mental model on model-based problem solving is *my* view on how others (professionals) look at the modelling process within their paradigm and associated practices.
[5] This decomposition will be discussed in section 6.4.3.
[6] 'Modelling archive' will be defined here as a 'set of modelling journals[7]'.
[7] 'Modelling journal' will be defined here as a 'structured record of modelling activities in a single modelling project, filed as instance of the underlying ontology'.

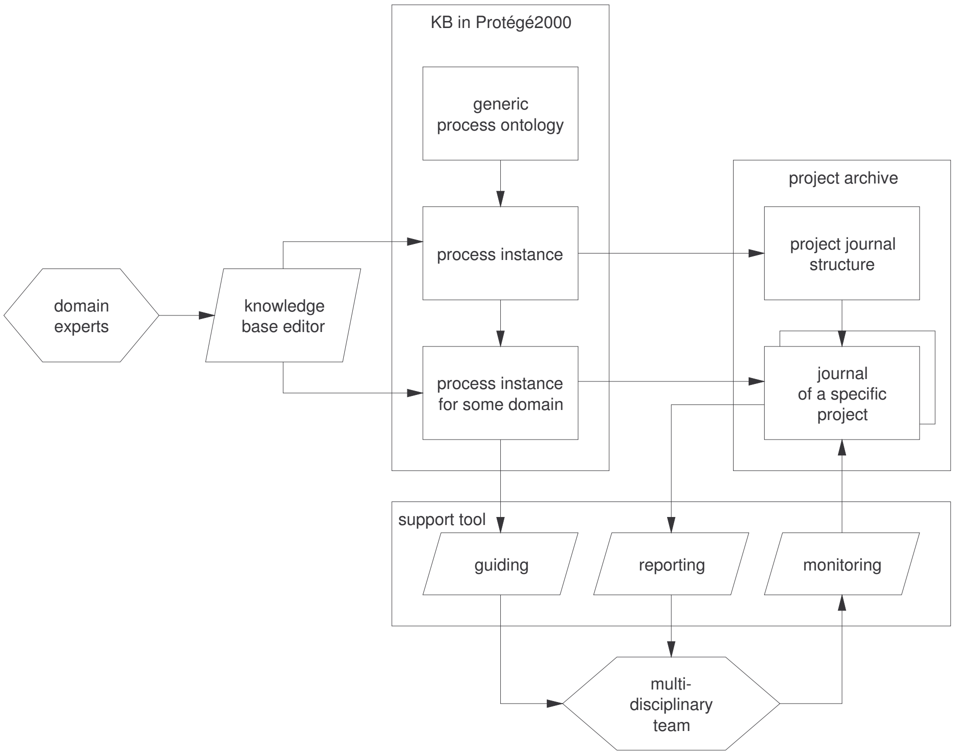


Figure 6-4. Sketch of the structure of the process KB and the major functions of the modelling support tool.

The design of the support system consisting of the basic required functions, guiding *what to do* and monitoring *what is done*, is based on many of the design aspects (6.4.4). In order to support collaboration within a team, the system should be Internet based and consist of a (thin) server application and a (fat) client application. The knowledge base with the modelling knowledge is situated at the server side and also the modelling journals, each associated with a single project. At the client side an application enables to download a local copy of the guidelines from the server. Furthermore the client application monitors what the modelling team members do and sends this information to the modelling journal associated with this model project on the server, to be used by all team members, which all have their own client application, but whose completed work is stored in a single modelling journal. The client application should also enable to produce reports on the work done, based on the information in the modelling journals.

6.4.2 The process of designing an ontological process knowledge base

A design of an ontological modelling knowledge base will be determined by many requirements, constraints and assumptions (Beulens and Scholten, 2000, 2001). Firstly, as said before, modelling will be viewed as a complex process, resembling to some extent business processes and fitting in the worldview of workflow management developers. Secondly, it is assumed that the complex process of modelling can be unraveled by decomposition into simpler sub-processes and these simpler ones into even more basic ones, until – in the lowest decomposition layer – *things-to-do* are atomic, i.e. so simple and unambiguous that they will be straightforward for the experts that do the modelling project. To some extent this procedure has been followed in the development of the GMP Handbook (Van Waveren *et al*., 1999, Scholten *et al*., 2000, Scholten, 2001, Scholten *et al*., 2001), but in the case of the GMP Handbook the decomposition of the whole process stopped after two decomposition levels. The GMP Handbook is actually an informal ontology and can be seen as a first step in developing a more formal ontology based modelling knowledge base. The decomposition and the selected granularity are discussed in section 6.4.3

To develop ontological knowledge bases one needs to select an ontological format and a well-equipped ontology development tool. The latter facilitates the development process significantly (in terms of design and of implementation). The choice of the format and the tool is interdependent and has its own selection criteria, which are discussed in Chapter 3.

The Good Modelling Practice Handbook, discussed in Chapter 2, aimed at realizing a knowledge base with modelling guidance based on a *highly informal ontology*[8]. The ontological knowledge base discussed in this chapter is more formal and should act as backbone of a system guiding modellers and support all persons involved in the modelling study at hand (Scholten and Osinga, 2001) The modelling support tool, depicted in Figure 6-3, should provide modelling guidance from the ontological knowledge base to a modelling team in a project and monitor and record what team members do. This leads to a series of design aspects, which will be discussed in section 6.4.4.

6.4.3 Knowledge decomposition and granularity

To decompose a complex process like the construction and utilization of models one needs criteria to indicate when and how an item (entity, concept) should be split up into its components. Zeigler and his colleagues (Zeigler *et al*., 1982, Rozenblit and Zeigler, 1986, Elzas, 1986, Elzas, 1989) developed a *multifaceted modelling methodology*[9], supported by a tool, ESP (Entity Structure Program). In their approach *entities* are the central issue and any decomposition of an *entity* should occur according to one or more decomposition criteria, called *aspects*. Their *aspects*, i.e. decomposition criteria, can be represented by ontological *relations*. The tool ESP was developed to model various facets[10] of systems. To some extend it resembles ontology development tools, but at present an ontological approach is preferred to structure and decompose pieces of knowledge[11].

For the development of an ontological knowledge base a decomposition strategy has been chosen that matches modelling perceptions and practices[12] of professionals. This perception is represented in the mental process model in Table 6-1. Item 4 in this mental process model (decomposition of the process) brings about two questions: (1) which *decomposition principles* can best be used and (2) what is a proper *granularity* of the decomposition?

To link these questions with design aspects, the following considerations can be used. Ontologies should consist of knowledge shared by a group. In the modelling instance of the process ontology (see subsections 6.6.3 and 6.6.4) a group consists of modellers and other team members involved in model-based problem solving. In line with this reason, intended users of the ontological KB should feel familiar with the guidance about modelling in the KB. The GMP Handbook is the result of a negotiation process within a group of modellers for water management and representatives of all stakeholders in the Netherlands (Van Waveren *et al*., 1999, Scholten *et al*., 2000, Scholten, 2001, Scholten *et al*., 2001). Therefore it is an adequate starting point for a set of decomposition principles. The decomposition in this GMP Handbook is restricted to two levels, but the description of what is to be done at the most decomposed of these levels allows a further detailing[13]. The actual principles, implicitly used in the GMP Handbook, are presented in Table 6-2.

Table 6-2. Decomposition principles, implicitly used in the GMP Handbook (Van Waveren, 1999).

Decomposition principles GMP Handbook
1. A decomposition should be practical, i.e. a *piece of work*[14] should not be decomposed into too many pieces of work;
2. A decomposition should not have too many levels, otherwise reading the text as a ‘book’ is difficult;
3. Each *piece of work* (a ‘concept’ in ontological terms) should fit in the perception of professional modellers; more specifically, all items should correspond with what professionals see as ‘parts’ of the

[8] Uschold and Gruninger (1996) define *degrees of formality of ontologies*, which is discussed in chapter 3.
[9] To this approach different references are made, including System Entity Structure, Multifaceted Structured Entity Modelling or MSE Modelling.
[10] *Facet* means here *aspect*, i.e. some set of *properties* of a system, often used in multi*facet*ed, i.e. showing different views, different *aspects*.
[11] The shift from the *multifaceted modelling methodology*, *System Entity Structure*, *Multifaceted Structured Entity Modelling* and *MSE Modelling* to the present ontological approach is rather incidental. The earlier approaches were embedded in the modelling community, while the ontological approach is based in the knowledge engineering community and related to internet.
[12] Professionals will typically have an expert view on their own modelling practices and not on other types of modelling (belonging to other modelling paradigms).
[13] Because the GMP Handbook (Van Waveren *et al*., 1999) is a book, further decomposition would have hindered its readability.
[14] *Piece of work* refers to a part of a process of an unknown size.

modelling process;
4. Each *piece of work* should be well defined and concise enough to be perceived by professionals as single pieces of work, functionally connected with each other, all together representing the whole modelling process.

For the present decomposition, associated with the process ontology in general and the process ontology for modelling in particular, these decomposition principles are still helpful, but additional requirements are needed too. GMP's second criterion (*not too many decomposition levels*) is important for books, but testing the modelling decomposition (described in Chapter 10) indicated that a modelling definition with too many decomposition levels is perceived as not very practical by professionals and confusing for novice modellers.

An additional requirement concerns the level of detailing, often called the granularity of the decomposition. The term granularity fits excellently in the ideas behind ontologies. If the granularity of the ontological knowledge base is too coarse, modelling will be described in too general, abstract terms. If the granularity is too fine, the risk of being too prescriptive will increase. Therefore a proper choice of the granularity is essential for the ontological knowledge base to develop properly and in a useable way. Table 6-3 presents a list of decomposition principles for the process ontology.

Table 6-3. Decomposition principles for the process ontology. These will be used as starting point for the design characteristics in section 6.4.1.

Decomposition principles process ontology
1. A decomposition should be practical, i.e. a *piece of work* should not be decomposed into too many *pieces of work*;
2. A decomposition should not have too many decomposition levels, in order to be of practical use;
3. Each *piece of work* (a 'concept' in ontological terms) should match with the perception of professional modellers and fit in their *performance repertoire*;
4. Each *piece of work* should be well defined and concise enough to be perceived by professionals as single *pieces of work*, functionally connected with each other, and all together representing the whole modelling process;
5. A single *piece of work* should have one or more clear inputs and one or more clear outputs;
6. The level of detailing should be sufficient in order to be concrete and small enough to be practical and acceptable for its intended users.

6.4.4 Design aspects of a process knowledge base and associated tools

As discussed in section 6.1, the ultimate goal is not to build an ontology, nor to build a process knowledge base (and populate it with modelling guidance). The ontological structure of the knowledge base should act as backbone of a system to provide help to its users throughout their projects, realizing collaborative processes such as modelling, and further support the work of all persons involved in those projects by developing a modelling support tool. This modelling support tool will be discussed in Chapter 7. Furthermore, it appeared cumbersome for domain experts to develop a KB using Protégé. This has been solved with a web based Knowledge Base Editor (KB-editor) that acts as a front-end for Protégé (see Figure 6-4). This tool is closely associated with the KB and will therefore be discussed in this section.

To realize an ontological process KB and to achieve an easy way to populate this KB, a number of requirements to the KB have to be considered. A requirement analysis consists of several parts:

1. elicitation of requirements;
2. analysis of requirements;
3. documentation of requirements;

The *elicitation of requirements* has followed a long path, starting with a basic idea (Scholten, 1994, Scholten and Udink ten Cate, 1999). Some of these ideas have been implemented in the GMP Handbook that was discussed in Chapter 2 (Van Waveren *et al.*, 1999, Scholten *et al.*, 2000, Scholten, 2001, Scholten *et al.*, 2001). The lessons learned formed the inspiration for the decomposition principles of Table 6-3 and partly also for a first set-up of the KB structure and the set of requirements in Table 6-4. This first set-up has been further used in a group of five modelling experts[15], who decomposed the modelling process itself. The results have been presented to and discussed by a lager group of 25 experienced modellers. These discussions resulted in the list of Table 6-4, although during the development process, the requirements have been adapted to new insights, based on experiences with intermediary versions of the KB. The list of Table 6-4 is the final version of the KB requirements and the result of stepwise development.

An *analysis of requirements* consists of order them in some classification. A typical classification for software requirements is described in Kotonya and Sommerville (1998). Their classification distinguishes *functional* (what the system should do) and *non-functional* requirements (constraints). The non-functional requirements concern (1) the process of software engineering, (2) the product to develop and (3) external requirements. Although often used, this classification of requirements is sometimes ambiguous. Glinz (2005) proposed a classification, in which four facets to requirements are distinguished: (1) *Kind*: including functional requirements, but also performance and others, (2) *Representation*: forms in which requirements are represented, (3) *Satisfaction*: requirements can be hard (i.e. yes or not fulfilled) or soft (i.e. gradual) and (4) *Role*: requirements play a role by (a) specifying properties of the system, (b) describing facts or rules of the system environment that influence the system and (c) specifying the behavior of actors in the system. For the requirements in this chapter the classification method of Glinz (2005) has been followed and the resulting classification is presented in detail in Appendix C

Documentation of the requirements can be found in Scholten and Osinga (2002). Here also the requirements to the modelling support (discussed in Chapter 7) and the test plan has been presented.

In Appendix C the requirements of Table 6-4 are classified according to the requirement classification of Glinz (2005) and translated in design solutions.

Table 6-4. Functional requirements to the KB and KB-editor.

Requirements
1. The KB should make knowledge explicit[16].
2. Have a proper granularity (i.e. level of detailing) of the knowledge in the KB.
3. KB should be flexible (i.e. easy to change its structure).
4. Be easy to maintain.
5. Be easy to update.
6. Adding and editing the KB should be protected by an adequate authorization system.
7. Some authority or board should control the content of the KB.
8. The KB should be 'open', allowing all interested persons to comment on the KB.
9. The part of the KB that has to be 'understood' by computers (the ontological structure) should be as formal as possible.
10. The part of the knowledge base that has to be understandable by human actors should be less formal and more textual.
11. The more basic ontological layers (layer 0, layer 1 and perhaps 2; see Figure 4-6 and requirement 21) of the ontological structure should be reusable for other processes.
12. The KB should be consistent.
13. The KB should be complete.
14. The knowledge should be shared by a substantial fraction of the professional community.
15. The KB and the support tool should reflect the work of (distributed) teams working in multidisciplinary model-based projects with different purpose types of different complexity.
16. The KB should enable diverse roles in the team.
17. The KB should enable diverse purpose types.
18. The KB should enable team members, from different domains and disciplines, to cooperate in synchronous and asynchronous subprojects.
19. The KB should enable different degrees of project complexity.
20. The KB should be suited for processes in general including, but not restricted to various types of simulation and mathematical models[17].

[15] The group had know-how on knowledge engineering techniques, experience in the development of modelling guidelines and software engineering skills. It consisted of Jens Christian Refsgaard (GEUS, DK), Hans Jørgen Henriksen (GEUS, DK), William G. Harrar (GEUS, DK), Ayalew Kassahun (WU, NL) and Huub Scholten (WU, NL).
[16] This is an evident requirement, as all knowledge bases make knowledge explicit.

21. The KB should be layered in a way that the most generic process knowledge is in a more generic ontological layer[18] (wit a lower number) and the most detailed and specialized knowledge in the more specialized ontological layers (with a higher number) or in instances of the ontology.
22. The developed KB and the KB Editor should be platform (operating system) independent.
23. The Knowledge Base Editor has to have to following functionality:
    - Set-up a new KB (supporting new types of processes);
    - Edit a KB which is developed based on the ontological structure described in this chapter; editing includes adding new aspects, changing, deleting, reading, commenting, etc.;
    - It should input the knowledge and its changes in the Protégé KB in the proper format;
    - There should an authorization system for a KB with the following types of users and privileges:
        - General users: may review and comment content of the KB;
        - KB editors: may edit the content of the KB;
        - KB administrators: may authorize users to edit and manage users and their privileges.
    - Version management of the KB;
    - Editing glossaries;
    - Registering users;
    - Enabling downloading KB and Modelling Support Tool.
24. The KB should be designed in way that its ontological format does not hinder moving to other formats.
25. The tool to develop the ontology should provide all basic functionality that can be expected from such a tool, including handling of names of concepts and relations, allocating identification numbers to all concepts and relations, and must harbor sufficient basic terminology.
26. Inexperienced knowledge experts should be able to upload their expertise to the KB.
27. The server should contain all shared information (i.e. KB with guidance and the work done by the team).
28. Adding and editing knowledge (i.e. guidance stored in the KB) should be carried out on the client side and stored on the server.

6.5 Second draft of the mental model on model-based problem solving

The mental model of model-based problem solving, introduced in Chapter 2 (Table 2-1) and updated in this chapter (Table 6-1), can now be extended, based on the design requirements[19]. The process of making the expertise of domain experts explicit (briefly discussed under *elicitation* in section 6.4.4) is not a part of this process model. The latter concerns only what teams have to do in problem solving using crisply defined terminology. The extensions of the mental process model include decomposition of the process and of projects instantiating the process.

Table 6-5. Second draft of the 'mental process model' summarizing how professionals view at multidisciplinary model-based problem solving. This is an update of the previous one in Table 6-1 and the first 6 items are copied from that table. Terms in *italic* are defined in the meta-ontology (Chapter 5); terms in **bold** are new terms, compared with the previous version of the mental process model. The present version will be extended in the next chapter.

Mental model of model-based problem solving, version 2

1. There is a *problem* to be solved, owned by some *problem owner*.
2. The *problem owner* organizes a *multidisciplinary team*, consisting of *team* members, who:
    a. Have different *disciplinary* backgrounds and
    b. Fulfill different *roles* in that *team*.
3. What *multidisciplinary teams* have to do ('solving' the problem at hand) is called a *process*, and is carried out in a *project* (in this chapter specifically a *modelling project*, instantiated for water management);
4. A *process* (a *composite action*) should be decomposed[20] into *single actions* that fit in the *performance repertoire* of *actors* involved in the *action*. A *process* should match the *accepted practices* of professionals in the field.

[17] 'Mathematical models' are defined in chapter 1 as: *models that share the use of mathematics to represent relevant parts of an object system to solve some problem(s) related to the object system. The models range from discrete or continuous simulation models to operation research models*.
[18] The *concept* '*ontological layer*' is discussed in the meta-ontology of Chapter 5; see also Figure 4-6.
[19] Actually the discussed solutions presented in Appendix C are also used.
[20] This decomposition is discussed in section 6.4.3.

5. *Practices* used in a *process* represent a body of (explicit and tacit) knowledge, often made partially explicit in textbooks and other publications, but typically not (fully) shared between disciplines involved in solving the problem;
6. *Processes* are often *multidisciplinary*, i.e. cover more than one *domain*.
7. The *project* consists of one or more **subprojects**, which belong to one or more *domains/disciplines*;
8. *Subprojects* are called:
   a. **Single domain** *subproject*, if they run with a different speed than other *subprojects*;
   b. **Multi-domain** *subproject*, if they are to be synchronized with other *subprojects*;
9. Each *subproject* consists of functional components, which means that they have a common purpose, a joint function; these functional components are the core pieces of work and will be referred to as **tasks**;
10. *Tasks* can be grouped for practical reasons (e.g. to provide more structure by adding an extra decomposition level; the groups of *tasks* will be called **steps**, because they typically have a sequential order, although with feedbacks, i.e. redoing *steps* or *tasks*;
11. Although *tasks* are functionally coherent, they consist of one or more **activities**;
12. To perform *tasks* and especially *activities*, *teams* or *team* members can use **methods** and **tools**;
13. The decomposition of a *process* can be stored in a **Knowledge Base** from which practical guidance will guide *teams* through the maze of things to do.

6.6 An ontological process KB

6.6.1 Introduction

The design aspects discussed in section 6.4.4 and the updated mental process model of section 6.5 permit outlining the design of the KB. As discussed before (see Chapter 4 and especially Figure 4-6 and requirement 21 in Table 6-4), this design will be structured in layers. In this section 6.6, I will start with some remarks on the ontological KB structure in general. Subsequently the design of the *ontological layer 1* (generic process knowledge; section 6.6.2) and *ontological layer 2* (specialized process instances with modelling knowledge; section 6.6.3) will be discussed. In section 6.6.4 the modelling knowledge of *ontological layer 2* will be specialized for simulation modelling knowledge in *ontological layer 3*. In section 6.6.5 the simulation modelling knowledge will be further specialized for water management (*ontological layer 4*) this ontological framework will be instantiated for simulation modelling for (multidisciplinary model-based) water management. *Ontological layer 5*, contains the modelling journal structure, as well as filled modelling journals in projects. General descriptions of the KB specialized for model-based water management and the KB-editor can be found in Kassahun *et al*. (2004), Kassahun and Scholten (2006), Scholten and Kassahun (2006), Scholten *et al*. (2004, 2006 and 2007).

Some remarks have to be made on the ontological structure. Presenting it as a structure suggests that it is static, but actually it is rather dynamic and since the first version has been proposed in September 2002, many changes have been made. The version presented here should therefore be considered as some intermediate 'frozen' state, designed and useful as a structure for the modelling knowledge base.

Protégé is a powerful tool, which supports developing ontologies in all stages. Just as other ontology tools it has some shortcomings. Firstly, Protégé has its own ontological jargon which does not always match with the ontological 'mainstream' opinions. To overcome this, a bootstrapping meta-ontology has been developed in Chapter 5 (especially Table 5.2). The second drawback is related to the first. Protégé requires that ontologies are represented in classes and subclasses. Furthermore, the term 'ontology' is not a well-defined concept. The tool should not determine how to define ontologies, but users should be facilitated to develop ontologies according to their own wishes. The last inconvenience of Protégé is its inadequate and poor graphical output. Therefore the structure of the modelling ontology, discussed in this chapter, has been redrawn from the structure developed with and stored in Protégé.

The layered structure of the process ontology as designed is illustrated in Figure 6-2. *Ontological layer 0* has been discussed in Chapter 5 and *ontological layer 5* (with the ontological structure for modelling project journals and instances of such journals) is discussed in Chapter 7. The *ontological layers 1*, *2, 3* and *4* will be discussed in this Chapter. The adequacy of the ontology as a structure for a (simulation) modelling knowledge base, for modelling in general and for processes, in which persons have to complete an operational process, will be evaluated in section 6.7 and especially in Chapter 10.

6.6.2 Ontological layer 1: generic process knowledge

Executing a *process* (defined in the meta-ontology of Chapter 5) in an actual situation is called a *project* (defined in the meta-ontology of Chapter 5). *Projects* in this book are typically *multidisciplinary*[21], which sets extra requirements to the ontological structure. To deal with these extra requirements, a *project* is divided in *subprojects*. The monodisciplinary parts of a *project* can be run more or less independently (asynchronously: own start and end time and own execution speed) or integrated in other parts belonging to one or more other disciplines (synchronously). The asynchronous parts of a project run in separate *subprojects* and the parts that integrate the work of more disciplines are executed in a common *subproject*. *Subprojects* can also be used for parts of a *project* dealing with issues that can better be run separate from other parts[22].

The most fundamental decomposition action divides a *subproject* in *tasks*. In the terminology of the 'Basic meta-ontology' of Chapter 5 it is still called a *composite action*), e.g. 'Describe Problem and Context'. Performing a *task* consists of doing one or more *activities*. A *task* is related to what has to be done and it refers therefore to the *process* to be executed in a *project*. An *activity* is the smallest piece of the *process*. Activities are associated with the role of a team member. *Tasks* and *activities* have often (but not always) one or more *methods*. A *method* is a scientifically based way of doing something or is some common (accepted) practice. Because the number of *tasks* in a *modelling project* is often rather high, an extra decomposition level between *process* and *tasks* has been introduced which will be called *step* henceforward. A *step* has no special meaning in the KB, in its ontology or in the support tool other than structuring the tasks in groups. This decomposition is depicted in Figure 6-5.

There are three versions of the concept *task*: *ordinary task* (abbreviated to *task*), *decision task*, from which one can continue to a next task or go back to one or more previous tasks and, finally, *review task*, which is a decision task, focused on the interactions between the roles in a team.

A *task* consists of one or more *activities* and associated *methods*, but to describe it, several *task elements* are used. These include (for each task) *name*, *formulation of definition*, *explanation of definition*, a list of *activities* and descriptions of each *activity*, a list of *methods* and description of each *method*, *sensitivity and pitfalls*, *references*, *software aspects*, *inputs*, *outputs*, *previous task*, *next task*, *receiving feedback from*, *providing feedback to*. Furthermore, many instances of these *task elements* contain hyperlinks to almost 1000 glossary terms. The definitions of the *concepts* of *ontological layer 1* are given in Appendix D and Figure 6-6 depicts the associated *structure diagram* of *ontological layer 1*. Together they represent the *generic process ontology*.

---

[21] *Multidisciplinary* is defined in the meta-ontology of Chapter 5.

[22] A typical example from model-based water management in a river basin: one *subproject* for all *socio-economic* issues and subprojects for each *subbasin* covering the following domains: *precipitation-runoff*, *groundwater*, *hydrodynamics surface water quality*.

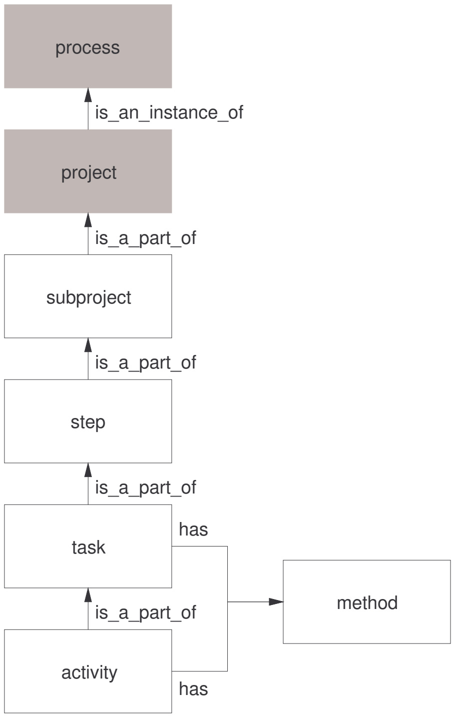


Figure 6-5. First draft of a structure diagram of the *generic process ontology* (*ontological layer 1*). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in the meta-ontology of Chapter 5.

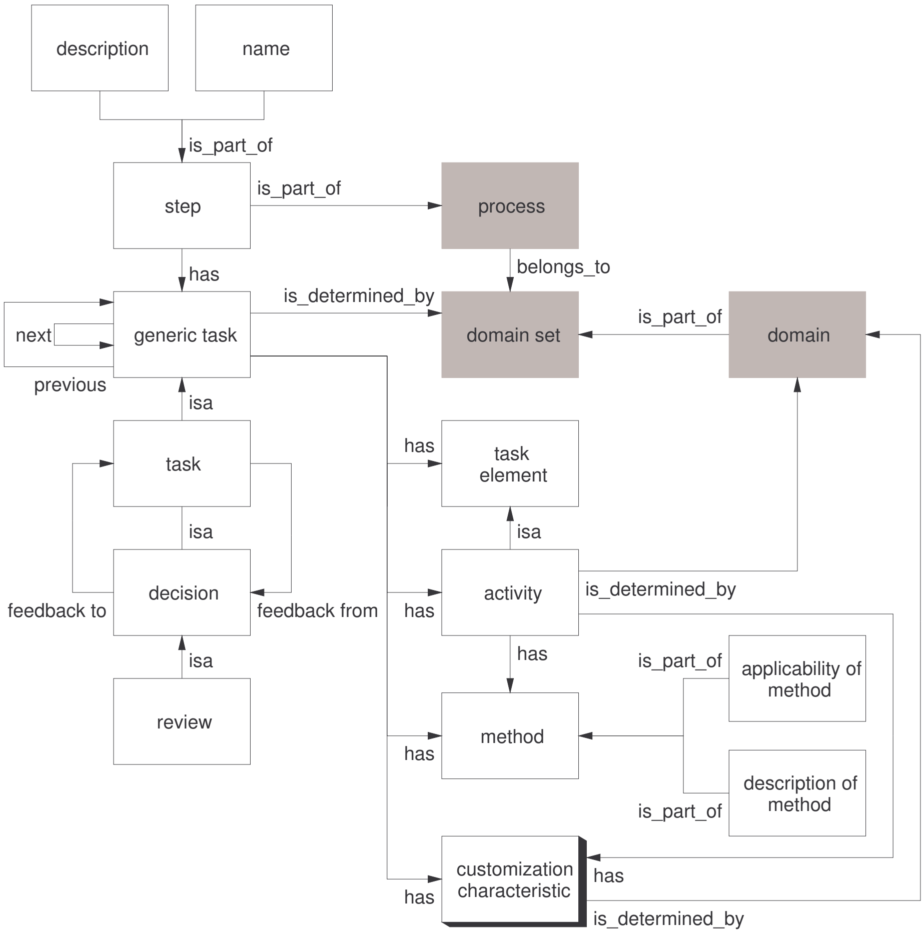


Figure 6-6. Detailed *structure diagram* of the *generic process ontology*. Rectangles are *concepts*, and arrows are *relations*. Grey *concepts* are defined in a less *specialized* ontological *layer*. Shadowed *concepts* are detailed in a more *specialized ontological layer* (section 6.6.3). The relation-type *isa* stands for 'is a …'. The *concept* '*customization characteristic'* will be detailed in Appendix D, Table D-1 and Figure 6-7. Many *concepts* and *properties* of *concepts* are not depicted in this figure.

The *concept* '*customization characteristic*', introduced in Figure 6-6, will be instantiated by presenting its structure in Figure 6-7. The *customization characteristic* '*domain*' represents the multidisciplinary aspects of the knowledge in the KB. Two *instances* of *domain*[23] (*generic domain* and *multi-domain*) are (rather) generic *concepts* and belong therefore to *ontological layer 1* (generic process ontology). The other *instances* of the *concept* '*domain*' are model-based water management related and belong therefore to *ontological layer 4*. The *instance* '*modeller'* of *concept* '*role'* is of *ontological layer 2* (specialized process ontology with modelling knowledge), but is still presented in this section, dealing with *ontological layer 1*.

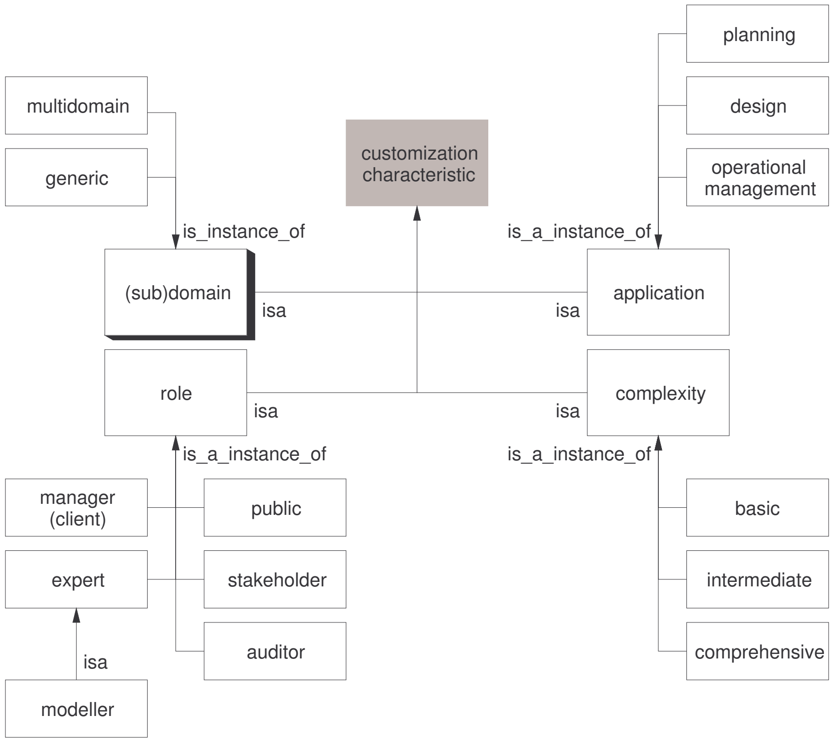


Figure 6-7. Extending the *structure diagram* of *ontological layer 1* (generic process ontology) for the *concept* '*customization characteristic'*, using the meta-ontology of Chapter 5 and the *generic process ontology* of 6.6.2. Rectangles are *concepts* and arrows are *relations*. Grey *concepts* are defined in a less *specialized* ontological *layer* i.e. the meta-ontology of Chapter 5. Shadowed *concepts* are detailed in Figure 6-11. The relation-type *isa* stands for 'is a …'. The concept modeller belongs to *ontological layer 2* (modelling knowledge; see section 6.6.3).

All definitions of the concepts in this section can be found in Appendix D, where Table D-1 defines the concepts of Figure 6-6 and Table D-2 those of Figure 6-7.

6.6.3 Ontological layer 2: modelling knowledge

In this section, the concepts belonging to the generic process knowledge of collaboration in multidisciplinary process are specialized for the process of modelling. In this book the modelling process is elaborated into more detail than the other two knowledge bases (problem-KB and model-KB), as a result of the HarmoniQuA project. At present, a new process-KB (with specialized knowledge) is set-up for the AquaStress project[24]. This project aims at water stress mitigation by providing guidance and tools to support this multidisciplinary process with strong emphasis on stakeholder participation. Although this process is also related to water management, it is completely different from multidisciplinary model-based problem solving for water management, as the latter is focused on using modelling (i.e. using mathematical models, e.g. simulation models) while the former is includes the water stress mitigation process.

Ontological layer 2 with modelling knowledge contains concepts relevant for modelling, but not belonging to a specific *modelling paradigm*. A part of the generic modelling terminology is based on the terminology

---

[23] *Domain* and *subdomain* are used here more or less as synonyms. In fact 'water management' has to be seen as *domain* and *specializations* of water management (e.g. *hydrodynamics*, *groundwater*) as *subdomains*, but in HarmoniQuA these *subdomains* are called *domains*.

[24] The MoST technology is used in the RTD, Integrated Project AquaStress, partly funded by EC DG Research, within the EU 6th Framework Programme, contract no. 511231-2.

provided by Refsgaard and Henriksen (2004), but adapted to make it more generic and less focused on simulation modelling for water management. Figure 6-8 depicts the (adapted) terminology of Refsgaard and Henriksen (2004) and this will be incorporated in the ontological framework in this book.

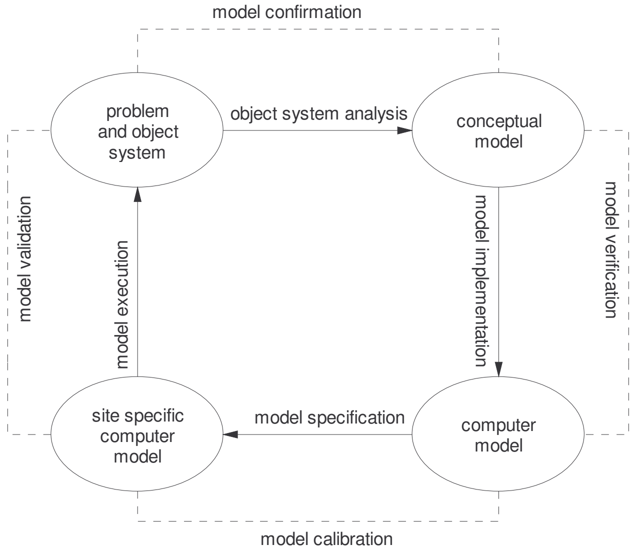


Figure 6-8. Elements of a modelling terminology (adapted from Refsgaard and Henriksen, 2004). The ovals represent what has to be modelled (object system and problem) and three forms of the model to solve the problem at hand. The inner arrows (solid lines) represent the composite actions to relate these and the outer arrows (dotted lines) represent the composite actions to evaluate the credibility of the inner arrows.

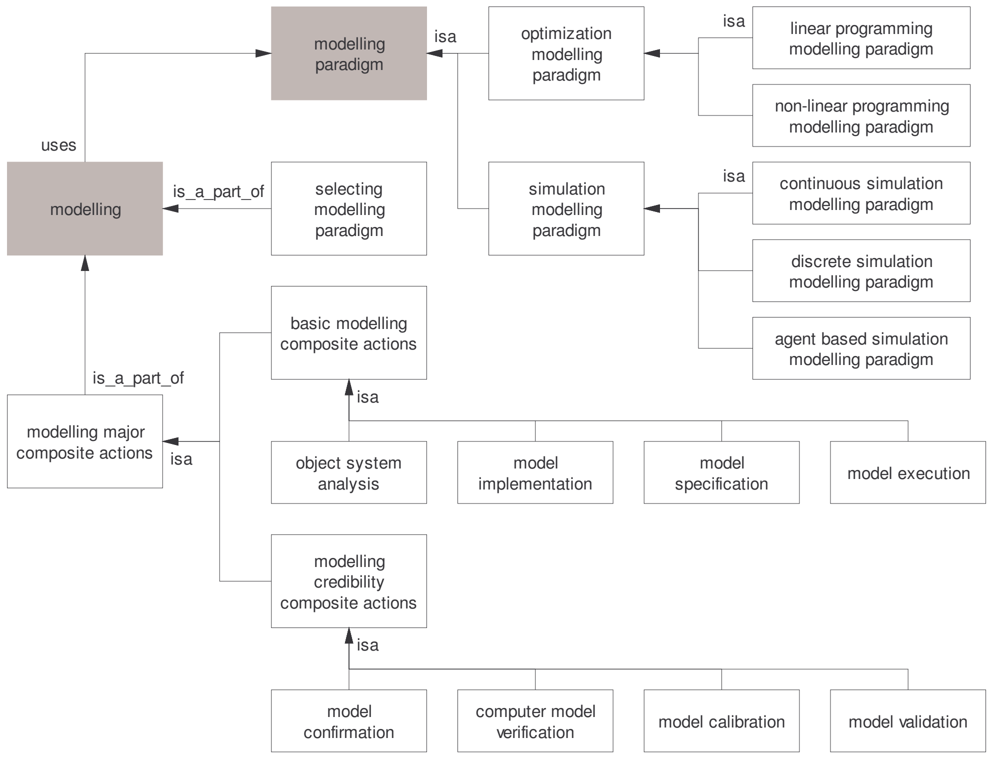


Figure 6-9. *Structure diagram* of the *modelling ontology*. Rectangles are *concepts*, ovals are *properties (of concepts)* and arrows are *relations*. Grey *concepts* are defined in a less *specialized* ontological *layer*. The relation-type *isa* stands for 'is a …' All concepts are defined in Appendix D, Table D-3.

How the concepts, related to modelling in general, are related is depicted in the structure diagram of Figure 6-9. In Appendix D, Table D-3 these concepts of *ontological layer 2* are defined.

A rather comprehensive overview of errors in *model implementation* checked in *computer model verification* is given in Ören *et al.*, 1985, but there are many more authors discussing this issue, including Oreskes *et al.*, 1994, Rykiel, 1996, Sargent, 1982, 1984a, 1984b and Sterman, 1991.

6.6.4 Ontological layer 3: simulation modelling knowledge

The main part of the *simulation modelling* knowledge in *ontological layer 3* of the process ontology consists of *instances* of the *concepts* '*step*' and '*task*'. These are defined *ontological layer 1* (see section 6.6.2) and are instantiated for the *simulation modelling* process (*ontological layer 3*) that is used for HarmoniQuA's *simulation modelling* KB. *Steps* will be presented with the *task elements* '*name*' and '*description*' and *tasks* by presenting the *task elements* '*name*', '*definition*' and '*explanation'* (if available). All other *task elements*, including *activities* and *methods*, are left out here, because presenting them here would request several hundreds of pages.

How these *instances* of *step* and *task* are related is depicted in Figure 6-10. Appendix D, Table D-3 gives the definitions and explanations of the *instances* of *step* and *task*. As far as they are included in this *ontological layer* 3, they belong to what is denoted as the *generic domain*. The complete textual version of the guidelines can be found at www.HarmoniQuA.org/public/Products/software.htm. In this way Figure 6-10 represents the flowchart (at *step* and *task* level) that *team members* have to follow in *simulation modelling* projects. This will be discussed in Chapter 7.

6.6.5 Ontological layer 4: simulation modelling knowledge for water management

Ontological layer 4 of the process ontology consists of two parts. The first part contains specializations of the water management *subdomains*. In *ontological layer 1* of the process ontology the concept '*domain*' is specialized into two *instances*, i.e. *generic domain* and *multidomain* (see Figure 6-7). Here, in *ontological layer 4* (simulation modelling knowledge for water management) a series of *subdomains* for model-based water management is introduced, depicted in Figure 6-11 and defined in Appendix D, Table D-5. These are actually used and implemented in the context of the HarmoniQuA project.

The second part consists of knowledge elements of *ontological layer 3* (simulation modelling knowledge), i.e. *instances* of the *concept* '*task*' that are not generic enough for *simulation modelling* in general, but that are restricted to specific *subdomains* of (model-based) water management. The *concept 'task*' is defined in *ontological layer 1* of the process ontology and instantiated for *simulation modelling* in *ontological layer 3*. The main part of the knowledge content of *tasks* consists of *activities* (see for the other knowledge containing *task elements*[25] section 6.6.2). If *tasks* are different for different water management *subdomains*, they differ especially in the content of the *activities* and not or hardly in *methods* and other *task elements*. When the process ontology with *simulation modelling* knowledge is used as guidance in water management projects (like in MoST, see Chapter 7), the *tasks elements* (especially *activities*) belonging to the generic *subdomain* are overwritten with the more specialized *simulation modelling* knowledge *tasks* for these *subdomains*, defined in *ontological layer 4*.

Two examples of *subdomain* specific *activities* are presented here. In the first case the content of a task is specified for specific subdomains by defining extra *activities* for one or more *subdomains*. An example of this approach is illustrated in Appendix D, Table D-6, which contains the activities for the task '*Summarize Conceptual Model and Assumptions*'. Next to the *activities* for the *generic subdomain*, two *subdomains* (*groundwater* and *socio-economics*) have a set of specific *tasks*, not relevant for the other *subdomains*.

In another approach *task* content is made *subdomain* specific by defining alternative *activities* for one or more *subdomains*. An example of this approach is illustrated in Appendix D, Table D-7 which contains the activities for the task '*Specify or Update Calibration and Validation Targets and Criteria*'. In the *activity* '*Select observation datasets*' alternative formulations are used for the *subdomains groundwater*, *precipitation-runoff*, *flood forecasting*, and *socio-economics*. The other *subdomains* (i.e. *biota*, *hydrodynamics* and *surface water quality*) have to follow the formulation of the *generic subdomain*.

The knowledge base for simulation modelling in water management distinguishes 48 *tasks*, which have in total hundreds of *activities*. Therefore, there are many of such examples, which are beyond the discussion here[26].

---

[25] The *task elements* are: *name*, *formulation of definition*, *explanation of definition*, *list of activities* , *descriptions of activity*, *list of methods* and *description of method*, *sensitivity and pitfalls*, *references*, *software aspects*, *input*, *output*, *previous task*, *next task*, *receiving feedback from*, *providing feedback to*.

[26] The filter function of MoST (see Chapter 7 and www.HarmoniQuA.org/tools) enables to see these differences.

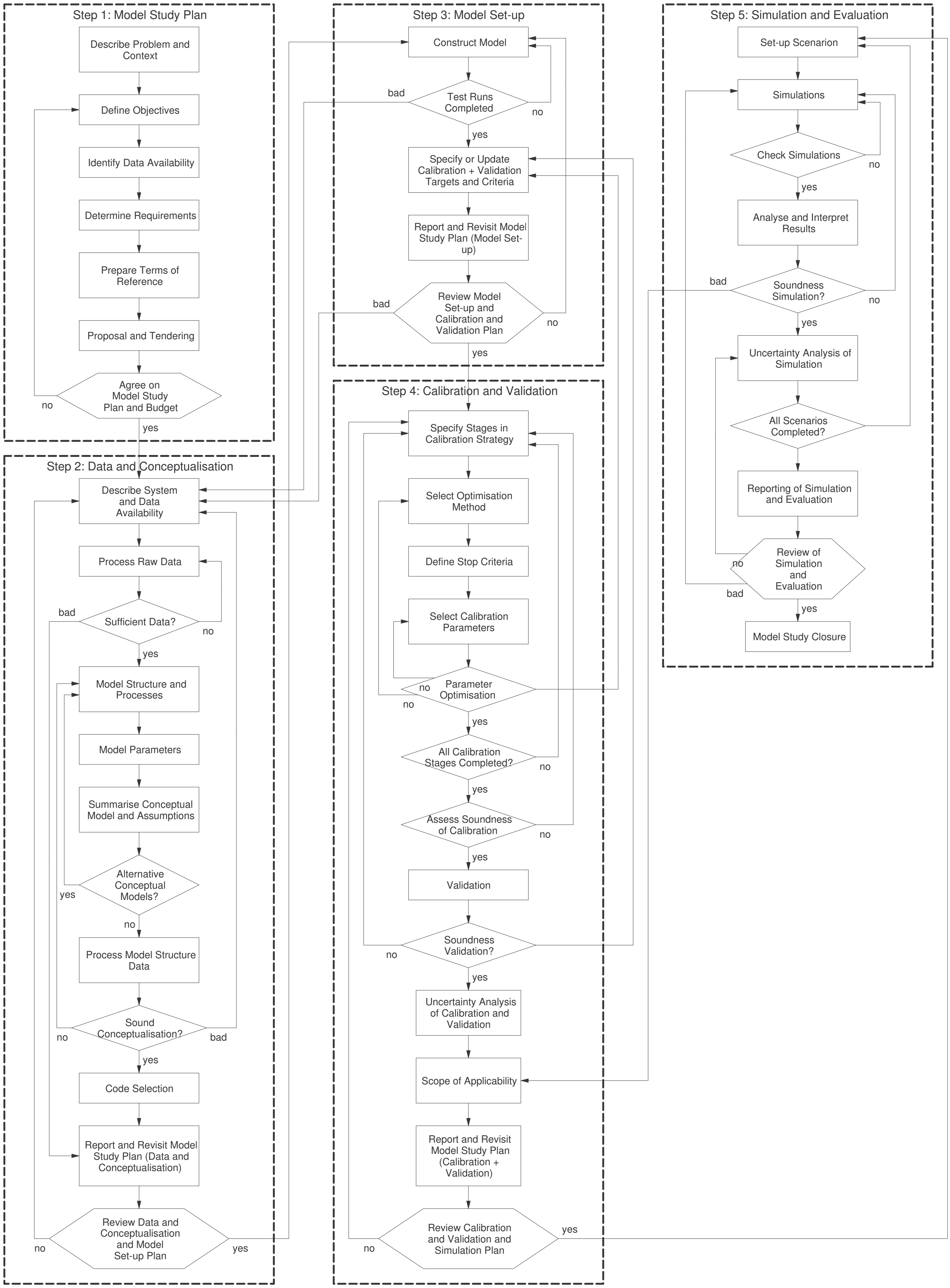


Figure 6-10. *Structure diagram* (*ontological layer 3*) of the *process ontology* containing simulation modelling knowledge with names of *steps* (dotted rectangles), *ordinary tasks* (rectangles), *decision tasks* (diamonds) and *review tasks* (hexagons). Arrows indicate relations of the type *next/previous*. (Scholten *et al.*, 2007 or see http://harmoniqua.wau.nl/Training/help/templates/available_templates.htm).

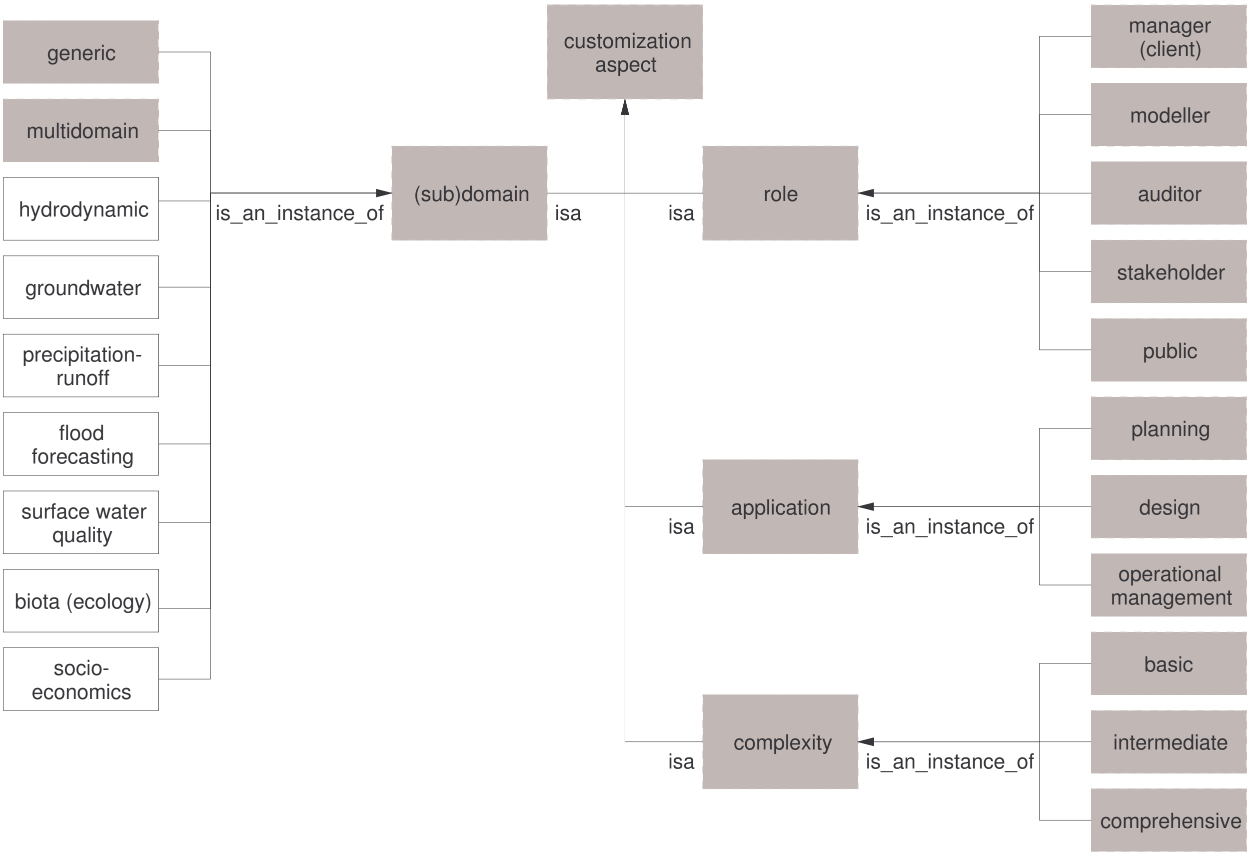


Figure 6-11. Specialization of the *structure diagram* for the modelling ontology (Figure 6-7), now for simulation modelling in water management (*ontological layer 3*), using the meta-ontology of Chapter 5, *ontological layers 1*, *2* and *3* of the *process ontology*. Rectangles are *concepts* and ovals are *properties* and arrows *relations*. Grey *concepts* are defined in a less *specialized* ontological *layer* or in *ontological layer 0* (meta-ontology of Chapter 5). The *relation*-type *isa* stands for 'is a …'.

6.7 The scope of the proposed KB

The KB has been designed, implemented and tested as to usefulness for *simulation modelling* in multidisciplinary model-based water management. Despite this design policy, the KB is designed in a way, which would allow extending it for other modelling paradigms or in other application areas, although this has been never demonstrated so far.

To assess the scope of the KB, three questions have to be answered: (1) can the KB be used for other modelling paradigms, (2) can the KB be used for other application domains and (3) can the KB be used for other processes?

The first question includes to which extent modelling-KB is useful outside *simulation modelling* for water management. Mathematical modelling in general implies other (mathematical) modelling paradigms, which means modelling with different types of solvers than just the differential equation type of solvers, typical for simulation modelling in water management and for many other application domains. At this stage this issue can not be resolved as using models belonging to other modelling paradigms, has not yet been attempted. It would require extending the modelling-KB with new *ontological layers 3*, *4* and *5*. Such a new *ontological layer 3* should contain modelling knowledge for another modelling paradigm than simulation, e.g. *linear programming*. Nevertheless, using the modelling-KB for other types of models and modelling seems promising, but it would require testing the idea in a case study context. There are no plans for such tests in a research setting.

Answering the second question (can the KB be used for modelling in other fields of application than water management?) seems simpler, as a large part of the KB (*ontological layers 1*, *2*, and *3*) has a generic character and the design of the knowledge base and its editor allow extending the KB to other *application* *domains*. The present (generic) modelling KB has been tested in four university courses[27], aiming at *simulation modelling* for other application domains, including modelling for forestry and nature conservation.

The third question can be summarized as follows. Can the KB be re-used for other processes than modelling? The term 'processes' has been used in this chapter earlier, but needs more explanation here. Not meant are the physical, chemical, ecological (other biological), economical or sociological processes, which are typically represented in *models* (discussed in Chapter 9), based on a defined *problem and associated object system* discussed in Chapter 8. The *processes* discussed in this Chapter 6 resemble more the business or research type of processes, in which persons have to collaborate in order to complete some complex job. Together with the Modelling Support Tool[28], discussed in Chapter 7) the KB can be used as a sort of workflow management system (see Chapter 2), but this requires another content of the KB, at least of the higher *ontological layer*s (*2*, 3 and especially *4*). The *ontological layer*s concept facilitates at least reuse of parts of the KB and this has been successfully tested in the project AquaStress, aiming at water stress mitigation. The reusability of the KB will further be discussed in Chapter 10 and in the final Chapter 11.

6.8 References

Beulens, A.J.M. and H. Scholten, 2000. A view on modeling paradigms: a prerequisite for Good Modelling Practice. Extended Abstracts of 14th JISR-IIASA Workshop on Methodologies and Tools for Complex Systems Modeling and Integrated Policy Assessment, July 3-5 2000, IIASA, Laxenburg, Austria.

Beulens, A.J.M., and H. Scholten. 2001. Model solving software and modeling paradigms: a challenge for developers. Pages 4-8: in M. Makowski, editor. Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, August 27-29, 2001, IIASA, Laxenburg, Austria.

Dourish, P. 2001. Process descriptions as organisational accounting devices: the dual use of workflow technologies. Pages 52-60 in Proceedings of the 2001 International ACM SIGROUP Conference on Supporting Group Work, October 2001. ACM, Boulder, Colorado.

Elzas, M.S., 1986. The applicability of artificial intelligence techniques to knowledge representation in modelling and simulation. In: M.S. Elzas, T.I. Ören, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era. Elsevier, Amsterdam, pp. 19-40.

Elzas, M.S., 1989. The role of artificial intelligence/entity structure concept manipulation methods in system modelling and design. In: M. Elzas, T. Ören, B. Zeigler (Eds.), Modelling and simulation methodology. Elsevier Science Publishers B.V., Amsterdam, New York, Oxford, Tokyo, pp. 3-27.

Glinz, M., 2005, Rethinking the Notion of Non-Functional Requirements, Proceedings of the Third World Congress for Software Quality (3WCSQ 2005), Munich (D), 55-64, http://www.ifi.unizh.ch/groups/req/ftp/papers/3WCSQ2005.pdf.

Kassahun, A., H. Scholten, G. Zompanakis and C. Gavardinas, 2004. Support for model based water management with the HarmoniQuA toolbox. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Proceedings of iEMSs 2004: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004, pp. 1282-1287.

Kassahun, A. and H. Scholten, 2006, A knowledge base system for multidisciplinary model-based water management, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006, ISBN 1-4243-0852-6 (978-1-4243-0852-1), http://www.iemss.org/iemss2006/sessions/all.html.

Kotonya, G. and I. Sommerville, 1998. Requirements Engineering: Processes and Techniques. Wiley, New York, ISBN: 0-471-97208-8, 294 pp.

Refsgaard, J.C. and H.J. Henriksen, 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.

[27] The modelling-KB has been used with MoST in the following courses at Wageningen University: INF-20306 *Programming and Modelling*, INF-31806 *Models for Forest and Nature Conservation*, GRS-30306 *Spatial Modelling* and INF-30806 *Advanced Modelling and Simulation*.

[28] The Modelling Support Tool, MoST, is for this reason extended to allow supporting other types of processes than just modelling (see. http://harmoniqua.wau.nl/public/Products/software.htm).

Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.

Rozenblit, J.W. and B.P. Zeigler, 1986. Entity-based structures for model and experimental frame construction. In: M.S. Elzas, T.I. Oren, B.P. Zeigler (Eds.), Modelling and simulation methodology in the artificial intelligence era. Elsevier Science Publishers BV (North-Holland), Amsterdam, pp. 79-100.

Sargent, R.G., 1982. Verification and validation of simulation models. In: F.E. Cellier (Ed.), Progress in modelling and simulation, vol., Academic Press, London, etc, pp. 159-169.

Sargent, R.G., 1984a. Simulation model validation. In: T.I. Oren, B.P. Zeigler, M.S. Elzas (Eds.), Simulation and model-based methodologies: an integrative view, NATO ASI Series F: Computer and Systems Sciences, vol., Springer Verlag, Berlin, etc., pp. 537-555.

Sargent, R.G., 1984b. A tutorial on verification and validation of simulation models. In: S. Sheppard, U. Pooch, D. Pegden (Eds.), Proceedings of the 1984 winter simulation conference, vol., pp. 115-121.

Scholten, H., 1994, Quality assurance of the simulation modeling process, A.R. Kaylan, A. Lehmann, T.I. Ören, Proc. of the European Simulation Symposium, Istanbul, October 9-12, 1994, Istanbul, Vol. I, 85-89.

Scholten, H. and A.J. Udink ten Cate, 1995. Testing simulation modelling quality, Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium, pp. II.A.3.1-6.

Scholten, H. and A.J. Udink ten Cate, 1996. Quality assessment of simulation modelling. In: C. Lokhorst, A.J. Udink ten Cate, A.A. Dijkhuizen (Eds.), Information and Communication Technology applications in Agriculture: state of the art and future perspectives. ICCTA, Wageningen, pp. 353-358.

Scholten, H. and A.J. Udink ten Cate, 1999. Quality assessment of the simulation modeling process. Computers and Electronics in Agriculture 22, 199-208.

Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, H. Wösten, R.D. Koeze and J.J. Noort, 2000. Good Modelling Practice in water management, Proceedings HydroInformatics2000 (on CD-ROM), vol., International Association for Hydraulic Research, Cedar Rapids, Iowa, USA.

Scholten, H., 2001. An ontological approach in quality assessment of modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering. TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0, Delft, The Netherlands.

Scholten, H. and S.A. Osinga, 2001. How to support GMP in model-based DSS. In: M. Makowski (Ed.), Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, August 27-29, 2001. IIASA, Laxenburg, Austria, pp. 45-47.

Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, J.H.M. Wösten, R.D. Koeze and J.J. Noort, 2001. Improving the quality of model-based decision support: Good Modelling Practice in water management. In: A. Schumann, J. Xia, M. Marino, D. Rosbjerg (Eds.), Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the International Association of Hydrological Sciences, Maastricht, The Netherlands, July 18-27, 2001, IAHS Red Books, vol. 268, IAHS, Maastricht, The Netherlands, ISBN, pp. 223-230.

Scholten, H. and S.A. Osinga (Eds.), 2002. Requirement analysis report of the HarmoniQuA knowledge base and tools. Wageningen University, Wageningen, HarmoniQuA-report D-WP2-1, 102 pp.

Scholten, H., J.C. Refsgaard and A. Kassahun, 2004. Structuring multidisciplinary knowledge for model based water management: the HarmoniQuA approach. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Proceedings of iEMSs 2004: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004, pp. 1288-1293.

Scholten, H. and A. Kassahun, 2006, Supporting multidisciplinary model-based water management projects: a user perspective, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006, ISBN 1-4243-0852-6 (978-1-4243-0852-1), http://www.iemss.org/iemss2006/sessions/all.html .

Scholten, H., A. Kassahun and J.C. Refsgaard, 2006, Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006, Research Publishing, Volume 3, 2231-2238.

Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.

Sterman, J.D., 1991. A skeptic's guide to computer models. In: G.O. Barney, *et al*. (Eds.), Managing a nation: the microcomputer software catalog, vol., Westview Press, Boulder, CO, USA, ISBN, pp. 209-229.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J. Noort, 1999. Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice. STOWA/RWS-RIZA, Utrecht/Lelystad, ISBN 90-5773-056-1, 149 pp.

Zeigler, B.P., D. Belogus and A. Boshoi, 1982. ESP- An Interactive Tool for System Structuring, Proc. of the 1980 European Meeting on Cybernetics and Research, Hemisphere Press.

7 A Modelling Support Tool for multidisciplinary model-based problem solving

There are two ways of constructing a software design;
one way is to make it so simple that there are obviously no deficiencies,
and the other way is to make it so complicated that there are no obvious deficiencies.
The first method is far more difficult.
(Charles Antony Richard Hoare, 1934)

Content of Chapter 7

This chapter is based on parts of the following papers:

MoST and KB-editor Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.

7.1 Introduction

Defining a process, such as multidisciplinary model-based problem solving for water management, does not guarantee, that teams going through the process will do what they should do. Achieving transparency of projects dealing with such a process requires more, e.g. recording what actually has been done. This chapter will describe a tool developed to support multidisciplinary model-based problem solving for water management. This support consists of guiding multidisciplinary teams through the modelling process, monitoring what teams have been doing and generating reports. In future versions, the tool should also provide advice based on previous, similar modelling projects.

Different aspects of the Modelling Support Tool alone or in combination with the KB with guidance on multidisciplinary model-based water management have been discussed also in Refsgaard and Henriksen (2004), Refsgaard *et al*. (2005) and Scholten *et al*. (2007). Refsgaard and Henriksen (2004) discuss existing guidelines and modelling terminology, which are mainly KB content related. Refsgaard *et al*. (2005) provide a classification scheme of modelling guidelines and provide the outline of some generic functionalities of the Modelling Support Tool. Scholten *et al*. (2007) review the Modelling Support Tool applied in the context of multidisciplinary, model-based water management.

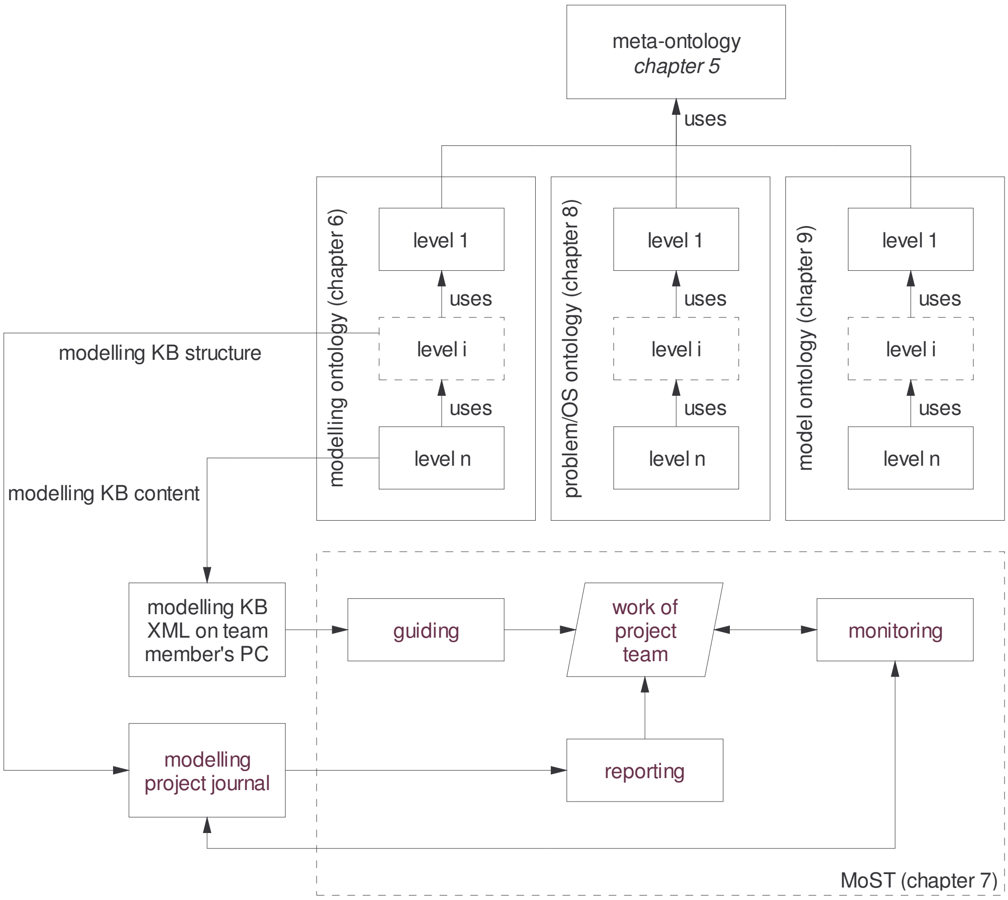


Figure 7-1. The ontological framework with the two dimensions (4 ontologies, each with *n* ontological layers of increasing specialization, of which 3 layers are depicted, i.e. *layer 1*, *layer I* and *layer n*..) See text for more explanation.

This chapter will have a different point of view on the Modelling Support Tool compared to Scholten *et al*. (2007), as this chapter will focus on the design process, design aspects (mainly functional requirements) and a detailed description of the translation of requirements into the Modelling Support Tool. Although this chapter aims at giving a more profound description of the Modelling Support Tool it will also be based on the realized version as implemented for the HarmoniQuA project.

7.2 Towards a Modelling Support Tool

7.2.1 The process of designing a Modelling Support Tool

The design of a Modelling Support Tool has followed a long and indirect route from a rough idea (Scholten, 1994, Scholten and Udink ten Cate, 1995, 1996, 1999), through the Good Modelling Practice approach (Van Waveren *et al*., 1999, Scholten *et al*., 2000, Scholten, 2001, Scholten and Osinga, 2001, Scholten *et al*., 2001) to the present Modelling Support Tool. Critical progress has been achieved in the HarmoniQuA project, in which requirements of such a tool have been formulated (Scholten and Osinga, 2002). This Requirements Analysis Report acted as initial outline for an incremental prototyping approach. This approach is similar to the one found in the software development method called Rapid Application Development (RAD). RAD has six core elements[1]: (1) prototyping, (2) iterative development, (3) time boxing, (4) limited number of experienced and flexible team members, (5) management approach, and (6) RAD tools. Although the used approach is not strictly RAD, the core elements of RAD will be explained here in the context of developing the Modelling Support Tool:

1. *Prototyping* is the rapid building of (light) functional or non-functional applications that can be used to discuss the functionalities with (future) users. Prototyping has been used intensively in the development of the Modelling Support Tool, mainly because it facilitates the discussion on functionality and layout between the group of developers and also with the end-users. Sometimes these prototypes were functional, but in the beginning they were just screens.
2. *Iterative development* consists of creating versions with increasing richness of features. In the development approach of the Modelling Support Tool, as proposed in HarmoniQuA's project proposal, three major development steps were been planned: a first (incomplete) prototype version, a full version and a final version. Because of the rhythm of the project (four years, general meetings with developers and end users twice a year and the tool development team an extra meeting in between the general meetings) more or less fourteen versions (i.e. the prototypes) have been developed[2].
3. *Time boxing* or the amount of features that can be realized in a fixed amount of time is an obvious choice, given the rhythm of projects. A primary disadvantage is the implicit mechanism to reduce the number of features to meet time box requirements. Still, this approach appeared productive and fitted seamlessly in the straitjacket of EC funded project contracts. A time boxing approach will guarantee a (working) version of the software under development, which is big advantage. In the development of the Modelling Support Tool the reduction in features – one of the main risks RAD – has hardly been noticed.[3]
4. *A limited number of experienced and flexible team members* has been realized in the HarmoniQuA project, where a few developers of two project partners were responsible for the development and a few persons of a third partner for testing the software[4].
5. The *management approach* following RAD should insist on short development cycles, enforce deadlines, help keeping high team motivation and focus on lowering bureaucratic hurdles. In HarmoniQuA I was the manager and the presented requirements to the management have been my starting points in managing the development of the Modelling Support Tool. Development cycles were short (2-3 months, which is longer than typical for RAD), deadlines have usually been met, team spirit was very high and bureaucratic hurdles could mostly be lowered, whether originating from EC headquarters in Brussels or from project partners.
6. *RAD tools* are necessary where speed is more important than cost. In the beginning of the project Delphi and Visual Basic have been used, next to proper version control software, while later in the project Java was the development platform. Delphi and Visual Basic are more appropriate for prototyping, while Java is more suited for a final product, as it is platform independent and stimulates best (Object Oriented) software engineering practices.

[1] See Wikipedia URL: http://en.wikipedia.org/wiki/Rapid_application_development.
[2] Fourteen is a rough estimate: 4 years with 2 general meetings plus 6 development team meetings in between represents 14 versions altogether.
[3] A note has to be made on the size of the time boxes: typical sizes of time boxes are 1 week to 1 month; the HarmoniQuA time boxes were 2-3 months on average.
[4] The typical size of the development group was four persons: all were involved in designing, one person involved in knowledge engineering aspects of the tool (related to the interfaces between the tool and the KB and between the tool and modelling journals, in which is recorded what teams do in projects), two software engineers and one manager guarding progress and requirements.

The three major releases (1.0 = *prototype version*, 2.0 = *full version* and 3.0 = *final version*) were official deliverables of the project and the intermediate releases have been used for testing and improving the software.

The software development approach followed here combines the advantages (and disadvantages) of RAD with the constraints imposed on international cooperation RTD[5] projects financed by the European Commission. These constraints include invariant resources and inflexible – time related – planning, which point to a RAD-like approach.

7.2.2 Design aspects of a Modelling Support Tool

The guidelines generated from the content of the process knowledge base discussed in the previous chapter, has a value in itself, but to use it, a Modelling Support Tool is needed that uses the ontological guidelines, guides the team through the network of things to do and keeps records of what is actually done. This section will outline the requirements for such a tool. In Appendix E the requirements of Table 7-1 are classified according to the requirement classification of Glinz (2005) and translated in design solutions.

Table 7-1. Functional requirements to the Modelling Support Tool, MoST.

Requirement
1. Modelling Support Tool[6] (MoST) should support multidisciplinary teams during the whole project lifecycle.
2. A GUI Menu should give access to all (most) functionality.
3. MoST should present the guidelines from the KB in various ways in order to support its use efficiently and effectively.
4. MoST should enable one to set-up projects.
5. MoST should monitor what team members do.
6. MoST should help making reports.
7. Glossary with terminology (per discipline) should be included.
8. The GUI of the tool should be easy to use without a steep learning curve, but it should also include most options wished by professionals that have to work with it on a daily basis.
9. The Modelling Support Tool should be portable to other operating systems (platforms).
10. The components of the tool (providing guidance, project initialization, project running, reporting) should be reliable, multipurpose, easy to maintain/update and fast.
11. The Modelling Support Tool should automatically use changes in the KB.
12. The Modelling Support Tool should be designed in such a way that it facilitates adapting it for other languages.
13. The Modelling Support Tool should be used by a distributed team working in a network setting (LAN or Internet).
14. The components of the system should be distributed in an efficient and effective network structure.
15. A single central server should be used for all versions of the KB in order to facilitate maintenance and upgrading.
16. Other servers should be used for projects.
17. Computers with connection to the central server and to the project server should run a client application for carrying out project work per team member.

The overall structure of the Modelling Support Tool, modelling KB and associated tools is depicted in Figure 7-2, including a central server, a project server and a client computer.

[5] RTD is Research and Technological Development.
[6] In fact, MoST is a generic Modelling Support Tool, but here associated with the KB for model-based water management. Furthermore, MoST is also used for AquaStress in a slightly improved version.

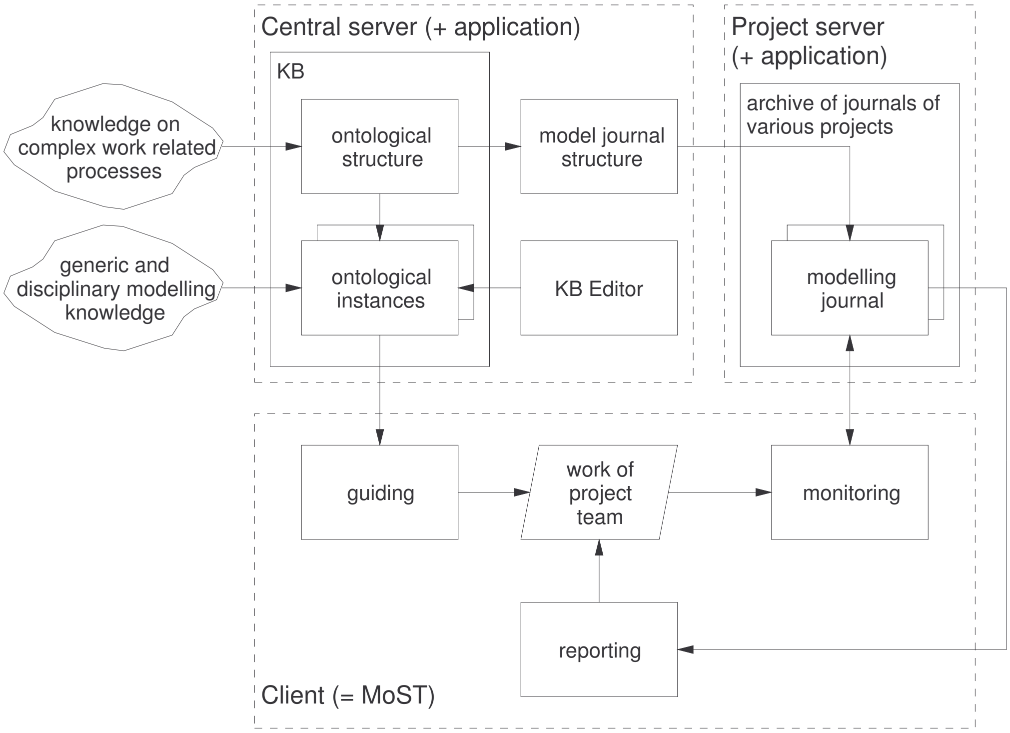


Figure 7-2. Sketch of the client-server architecture with MoST's process knowledge base and its major functions, being initializing projects, guiding, monitoring and reporting. The process guidelines are stored in the ontological instances. Journals of specific projects are located on a project server, being a LAN (team projects, in which teams are within a single organization) or some controlled Internet server (team project with distributed team members). Adapted from Scholten et al., 2006.

7.3 Third draft of the mental model on model-based problem solving

The mental model for process support, introduced in Chapter 6, section 6.5, can now be extended, based on the design requirements section 7.2.2 and the solutions presented in Appendix E. The process of making the expertise of domain experts explicit is not a part of this process model. The latter concerns only what teams have to do in problem solving. The extensions of the mental process model include decomposition of the process and projects instantiating the process[7].

The mental modelling process developed in two stages in the Chapter 6, has to be extended. The first 13 items are introduced in the first two versions of the model (see Chapter 6).

Table 7-2. Third draft of the 'mental process model' summarizing how professionals view at multidisciplinary model-based problem solving. This is an update of the previous one in Chapter 6, section 6.5 and the first 13 items are copied from that table. Terms in *italics* are defined in the meta-ontology (Chapter 5); terms in **bold** are new terms, compared with the previous version of the mental process model.

Mental model on model-based problem solving, version 3

1. There is a *problem* to be solved, owned by some *problem owner*.
2. The *problem owner* organizes a *multidisciplinary team*, consisting of *team* members, who:
   a. Have different *disciplinary* backgrounds and
   b. Fulfill different *roles* in that *team*.
3. What *multidisciplinary teams* have to do ('solving' the problem at hand) is called a *process*, and is carried out in a *project* (in this chapter specifically a *modelling project*, instantiated for water management);
4. The *process* (a *composite action*) should be decomposed[8] into *single actions* that fit in the *performance repertoire* of *actors* involved in the *action*. This *process* should match with *accepted practices* of professionals in the field.

[7] A *project* is defined in the meta-ontology (Chapter 5) as: planned *process*; running a *process* in a *project* makes the *project* an *instance* of the *process*.
[8] This decomposition is discussed in Chapter 6, section 6.4.3.

5. The *practices* used in the *process* represent a body of (explicit and tacit) knowledge, often made partially explicit in textbooks and other publications, but typically not shared between disciplines involved in solving the problem;
6. *Processes* are often *multidisciplinary*, i.e. cover more than one *domain*.
7. The *project* consists of one or more *subprojects*, which belong to one or more *domains/disciplines*;
8. *Subprojects* are called:
   a. *Single domain subproject*, if they run with a different speed than other *subprojects*;
   b. *Multidomain subproject*, if they are to be synchronized with other *subprojects*;
9. Each *subproject* consists of functional components, which means that they have a common purpose, a joint function; these functional components are the core pieces of work and will be referred to as *tasks*;
10. *Tasks* can be grouped for practical reasons (e.g. to provide more structure by adding an extra decomposition level; the groups of *tasks* will be called *steps*, because they typically have a sequential order, although with feedbacks, i.e. redoing *steps* or *tasks*;
11. Although *tasks* are functionally coherent, they consist of one or more *activities*;
12. To perform *tasks* and especially *activities*, *teams* or *team* members can use *methods* and *tools*;
13. The decomposition of a *process* can be stored in a Knowledge Base from which practical guidance will guide *teams* through the maze of things to do.
14. A **Modelling Support Tool** should support *teams:*
    in the **project definition phase** by helping to **set-up projects** by:
    a. Composing **teams** and specifying associated **authorizations**,
    b. Defining *subprojects* (**selection of domains** and **selection of relevant tasks** from templates);
    c. Editing **auditing criteria** (based on a default set);
    and in the **project running phase** by:
    a. **Providing guidance** from the *Knowledge Base* and presenting it to team members with different **views** on a process (**tree view**, **flowchart view**, **task view**), **customized guidance** for domains, roles of team members and with **project templates** (predefined and to be defined by a user);
    b. **Monitoring** what team members do in a project;
    c. **Reporting** for various audiences and purposes.
15. A **Client-server architecture** should enable cooperation in (multidisciplinary) teams in the following way:
    a. The KB can best be made accessible on a **central server** with a **server application**;
    b. Journals can best be stored on a **project server**;
    c. The Modelling Support Tool, MoST is the **client application**.

7.4 The Modelling Support Tool MoST

7.4.1 Introduction

Based on sections 7.2.2 and Appendix E the Modelling Support Tool can now be described. It consists of a component to set-up and edit projects (section 7.4.3), getting guidance (section 7.4.2), monitoring what teams actually do in a project (section 7.4.5) and reporting what has to be done (section 7.4.6). Subsequently some other features – not yet implemented – will be discussed (section 7.4.7). Detailed descriptions of the tool and its usefulness for model-based water management are given in Scholten *et al*. (2004 and 2007).

The relation between the Modelling Support Tool and the other components of the system are depicted in Figure 7-3 with emphasis on how the components interact with each other and the role of team members and domain experts in this system. The next section will detail the functionality of the Modelling Support Tool in a project.

Depending how the tool is started, users will enter through the **guideline component** (if no project is opened) or through the **project component** (by double clicking on a modelling journal[9]). In the former case the project component tab and the **reporting component** tab will not be visible, in contrast with the latter case (see Figure 7-6 and Figure 7-7). In section 7.4.2 the *guideline* component will be presented, in

[9] In MoST's successor ProST called project journal.

section 7.4.3 *initializing projects*, in section 7.4.5 *running projects* and in section 7.4.6 *reporting*. The section will conclude with *other functionality* (section 7.4.7).

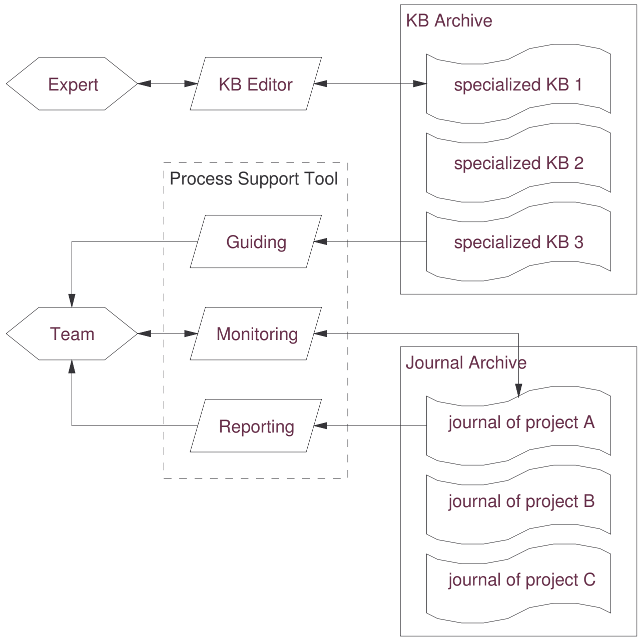


Figure 7-3. The overall modelling support system consists of KB Archive (with several Knowledge Bases), KB Editor, Modelling Support Tool (MoST for guiding, monitoring and reporting) and Journal Archive (upper right corner). *Guiding*, *monitoring* and *reporting* are discussed in sections 7.4.2, 7.4.5 and 7.4.6 respectively.

7.4.2 Guideline

The guideline component provides three views on the KB: the tree view, the flowchart view and the task view. These views are the major panels on the screen in the guideline component (Figure 7-4 and Figure 7-5).

The **tree view** resembles standard windows trees[10] like in windows explorer and enables one to browse through the steps and tasks in a similar way as Windows allows browsing through directories and files. The '+' sign will allow to expand a *step* into *tasks* and the '-' sign will collapse *tasks* into *steps* (see Figure 7-4 and Figure 7-5). Four symbols indicate the nature of the *concepts*:

1. Red ovals for *steps*;
2. Yellow rectangles for (*ordinary) tasks*;
3. Blue diamonds for *decision tasks*;
4. Green hexagons for *review tasks*.

The **flowchart view** provides another type of browsing through the network of *steps* and *tasks*, but also gives a structured view with the order of *tasks* and feedback loops. Arrows indicate the order and have tool tips to see their origin *task* or destination *task*.

The task view: presents a textual description of the chosen concept (*guideline*, *step* or *task*). It shows the content of the task. If a task has been selected, several tabs can show specific parts of its content. The descriptions of tasks in this view contain discipline related terminology, which is explained in the glossary. This glossary can be accessed through the hyperlinks in the task view or through the menu (*help / glossary…*).

[10] A *tree view* is a graphical diagram used to display the hierarchal structure of items, such as directories and files on a disk.

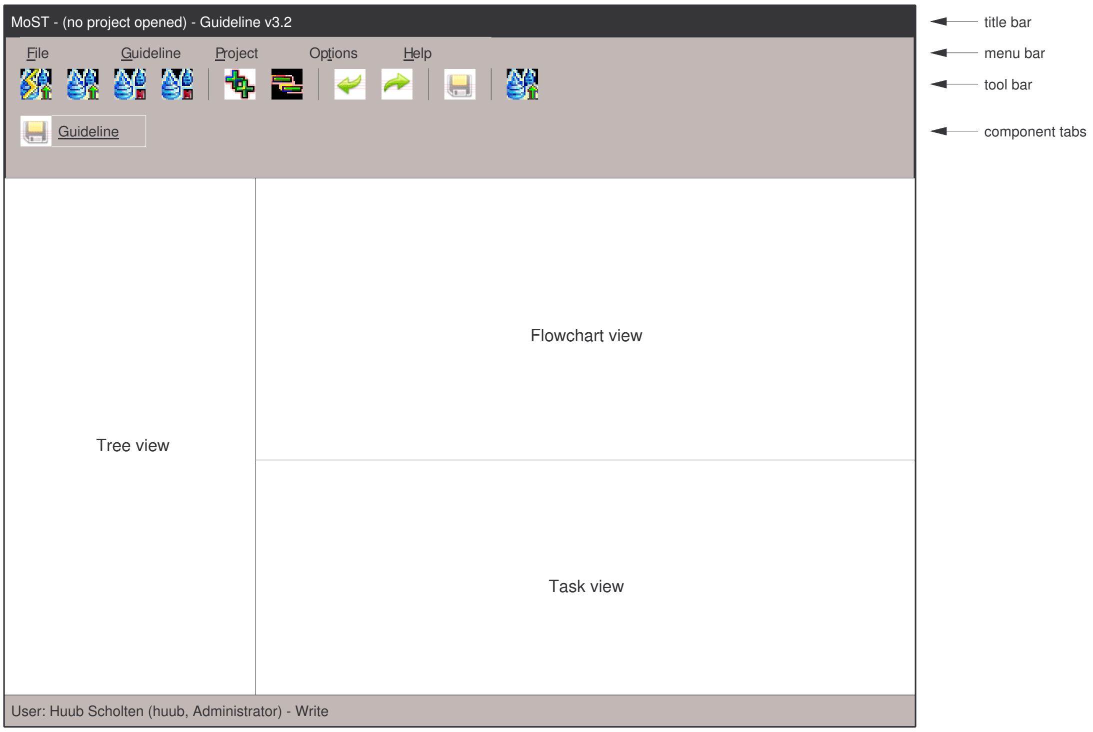


Figure 7-4. Schematized layout of MoST GUI in case no project has been opened with menu bar, tool bar and guideline component tab in the top and the tree view, the flowchart view and the task view in the central part of the screen.

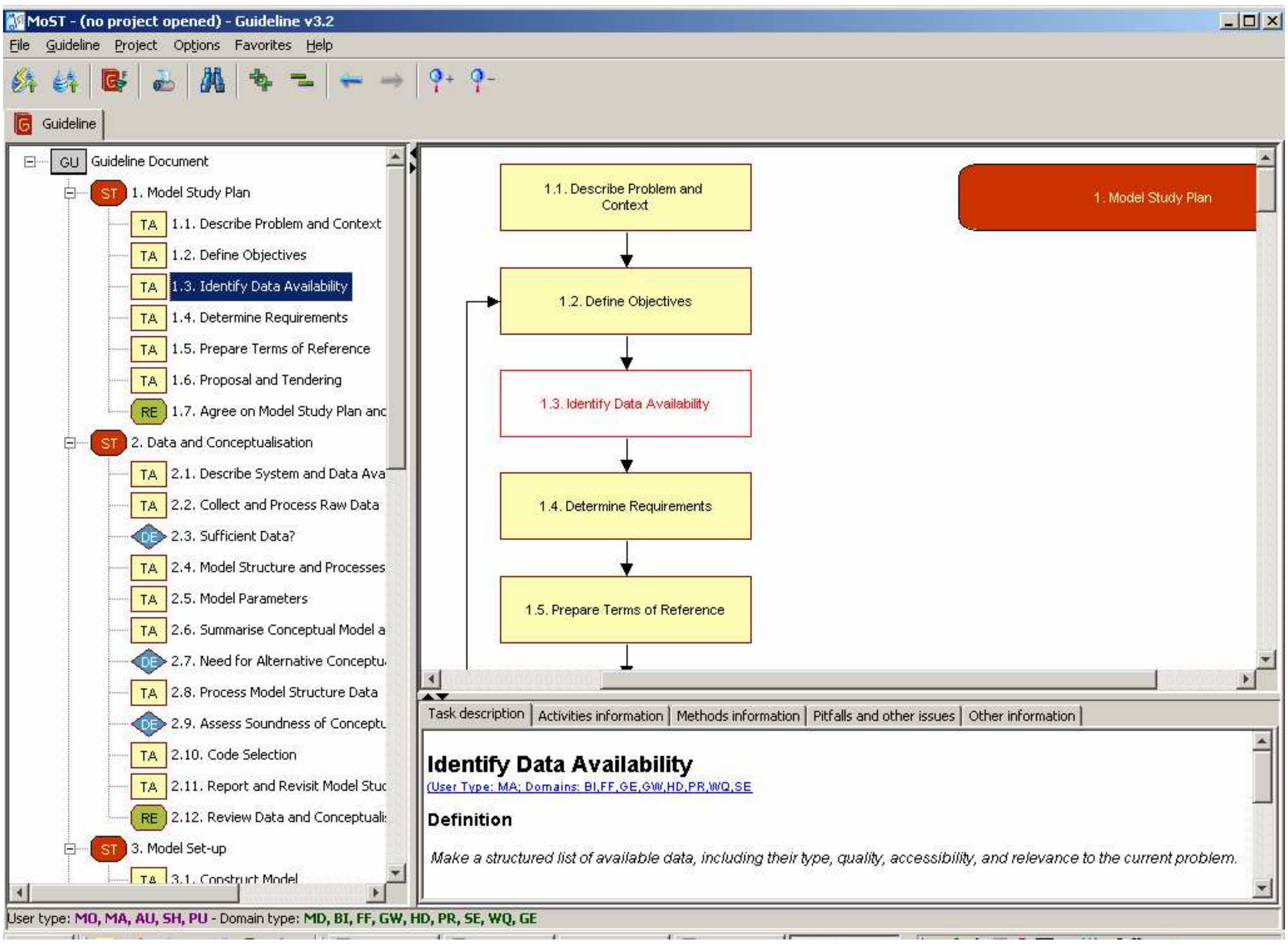


Figure 7-5. Screen dump of guideline component. In the tree view task 1.3 has been chosen and the guideline component synchronizes the other views by setting the pointer to the same activity in these views.

One of the major aims of the overall system is customizing the guidelines (and what team members have to do) to their roles, the domains / disciplines involved, the complexity of the case and its purpose. Therefore the guideline component of the tool allows its users to filter on basis of these customization properties[11]. Filtering will automatically update all the *tree view* and the *flowchart view*.

The menu item *guideline* also enables several functionalities, including selecting a guideline version (from local computer), downloading guidelines (from central server), printing the flowchart and printing the text of the active task[12]. The guidelines menu item also allows finding terms, expanding and collapsing the tree and zooming-in or –out in the flowchart, adapting fonts and colors.

The guideline component can also be used outside the context of a project.

7.4.3 Running projects with the Modelling Support Tool

There are several ways to start working on a project

1. For a new project:
    a. *New Online Project*: a new project has to be defined, which will be stored on a central server[13] to allow distributed teams to co-operate.
    b. *New Local Project*: a new project has to be defined, which will be stored on a local hard disk; this approach is typical for single person projects.
    c. *New Project from Template*[14]:
        i. *Local*: a new local project, but a part of the definition is included in the template (relevant subproject(s), relevant tasks, (optional) users, (optional) authorizations of users).
        ii. *Online*: a new online project, but a part of the definition is included in the template (relevant subproject(s), relevant tasks, (optional) users, (optional) authorizations of users).
2. For an existing project
    a. *Local project*: there are 2 options:
        i. Double click on the modelling journal file (*filename*.mpj);
        ii. Start MoST and use (in the menu) *Open Local Project*;
    b. *Online project*: Start MoST and use (in the menu) *Open Online Project*;

The choice for a *local* or for an *online* project is not permanent, as one can switch between local and online, but there are consequences. When working in a local project, only one user can work on a project. Imagine, that a multi-user online project is transformed to a local project (menu: *Project/Export online Project to Local*) from that moment on, only one user can work on the project. If others continue to work on the online version, there will be a synchronization problem, as the online version is permanently synchronized when a user changes the content of the modelling journal. The independently edited local version cannot automatically be synchronized with the online one. Starting with a single member team on a local project is less tricky, as no others can participate unless on the same local version on the same computer. After transforming a local project to an online one (menu: *Project/Convert Local Project to Online*), synchronization is guaranteed, as long as it stays online.

Starting a project from a template is straightforward. Templates can be found on the hard disk in a directory called *templates*, which is a subdirectory of the directory where MoST has been installed. So far several templates are available for specific types of multidisciplinary model-based water management projects[15]. Other templates can be generated from (successful) projects (menu: *File/Save the Project as Template*), allowing to include all users and their authorizations (efficient in routine type of projects) in team running the project or to exclude team member information.

---

[11] At present *job complexity* and *application purpose*, although implemented in the tool, have been switched off, because these options were not used in modelling test cases of HarmoniQuA.
[12] To print the full guideline it can be downloaded for printing from www.harmoniqua.org/tools.
[13] A central server can be accessible through Internet or some LAN-server within (a department of) an organization. An example of a central server for model projects is http://harmoniqua.org/tools.
[14] Project *templates* are empty modelling journals containing the relevant structure of a project, i.e.: relevant subproject(s), relevant tasks, (optional) users, (optional) authorizations of users.
[15] See http://harmoniqua.wau.nl/training/help/templates/available_templates/available_templates.htm.

7.4.4 Initializing projects

Projects have to be defined in the project initialization phase. In a *window with tabs*[16] the project initiator has to specify the project to start. After giving the project a name, the following *tabs* have to be used for setting up a project:

- **Subprojects**: selecting one or more *domains* from a list of available *domains* and adding this set to the list of *subprojects* and replacing the default *subproject* name by a meaningful one defines a *subproject*.
- **Tasks**: For each *subproject* one can deselect tasks from the full list or, if used a template project, select and deselect *tasks*. No *tasks* can be defined here. One can also choose between random access of the *tasks* to do or enforce strictly following the order of *tasks* given by the guidelines.
- **Users**: Users[17] are added by providing username, password, first name and last name, selecting one or more roles per user and adding the user to the user list.
- **Authorization**: For each user the project initiator has to specify authorization. First one has to decide whether to disable or enable login account and if the user is a project administrator. The project initiator is by default administrator, unless this authorization is deselected. Furthermore per user and per subproject the following privileges: *write*, *decision making* (allowed to take decision in decision tasks of that subproject) and *reading* of material per role.
- **Scoreboard**: Project administrators are allowed to add or remove questions in scoreboards. A list of default scoreboard question is provided by default. Managers[18] (i.e. the clients of the project) are allowed to edit weights of scoreboard questions and during projects auditors can use the scoreboards in their evaluation activities.

Projects can have more than one project administrator. Administrators have several privileges, all related to smooth the progress of the project and to assist team members in their daily practice. Most of these privileges concern project settings (i.e. project specifications specified at project initialization).

7.4.5 Running projects

Keeping records is essential in the project running phase, as it enables doing audits and allows reconstruction of projects. This monitoring component of MoST is complex in its user interface and in its functionality.

The project component provides three views of which two on the KB: the tree view and the task view. The third view, the activity view, presents. These views are the major panels on the screen in the project component (Figure 7-6 and Figure 7-7).

The **tree view** (left in Figure 7-6 and Figure 7-7) displays all the tasks in the project and the status of these tasks. This view can be used to browse through the necessary tasks and also to inspect the status of each task in the subproject. Different icons are used to show this status. A transparent rectangle indicates a task not yet started. A yellow rectangle indicates a task that has already started, but not yet finished. If it is finished, a green checkmark is displayed. Skipped task are shown with a red cross.

The **task view** (lower right in Figure 7-6 and Figure 7-7) is similar to the task view in the guideline component of MoST (section 7.4.2). It provides guidance on the task that is selected in the tree view, with tabs to select the sections on the guidance for that task.

---

[16] In MS Windows terminology it is called a *tab control*.
[17] *Users* are team members.
[18] In MoST, the implementation of the Modelling Support Tool for HarmoniQuA, called ‘Water manager’.

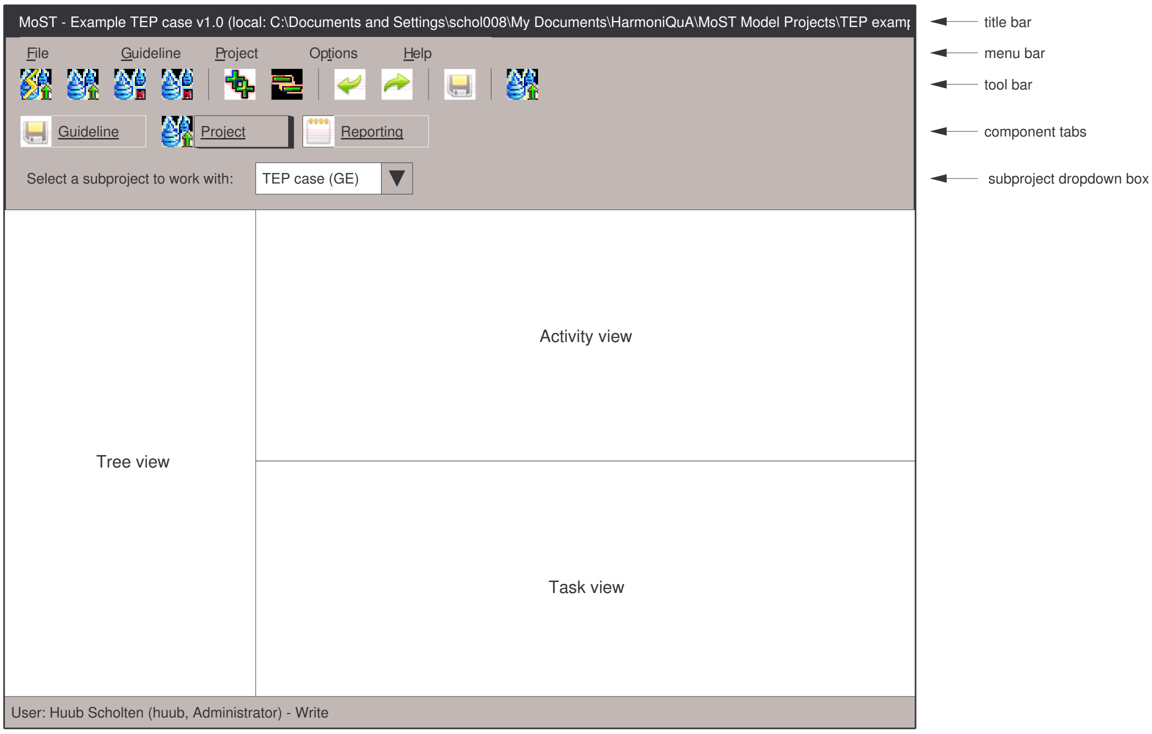


Figure 7-6. Schematized layout of MoST GUI in case a project (TEP case (GE)) has been opened with menu bar, tool bar, component tabs and subproject dropdown box. In the top and the tree view, task view and activity view in the central part of the screen.

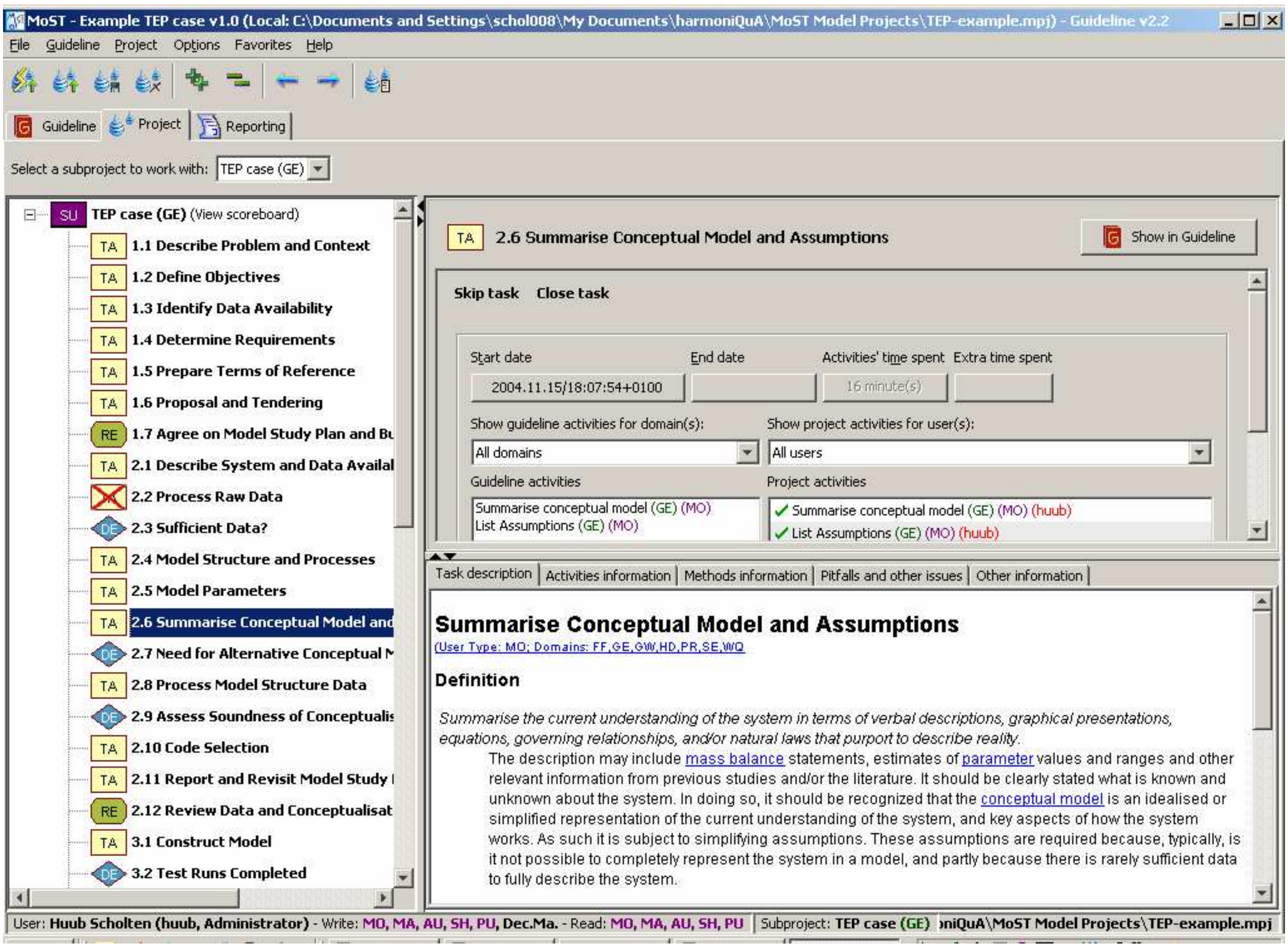


Figure 7-7. Screen dump of MoST GUI, see Figure 7-6 for explanation.

The **activity view** (upper right panel in Figure 7-6 and Figure 7-7) is the main panel to record what team members do. What team members do in this view will be recorded in a process journal (e.g. in case of modelling a model journal). After selecting a *task* in the *tree view*, there are three possibilities:

1. **Open at Task level**: all activities of the task at hand are handled as a single activity;
    a. Menu options:
        i. **Skip task**: see below;
        ii. **Close task**: when completed a task can be closed, prohibiting that users continue to work on that task; only project administrators can re-open it;
        iii. **Skip activity**: similar to skip task, see below;
        iv. **Finish activity**: when the combined *activities* of the *task* at hand are finished, a small window pops-up showing the time between start and end of the combined *activities* and allowing to fill-in time spent;
        v. **Manage time**: a small window pops-up showing the time between start and end of the combined *activities* and allowing to fill-in time spent;
        vi. **Work at Activity level**: switch from *task level* to *activity level*;
    b. Provide *detail, actions and outcomes* of the combined *activities* of that *task*;
    c. Select *methods* used;
    d. Allow to attach and retrieve files to the modelling journal, accessible from the task at hand;
    e. In case of a *decision task* and *review task*, decide which is the following *task* to do (i.e. going back to redo a previous *task* or continue with the next one).
2. **Open at Activity level**: all *activities* of the *task* at hand are handled as separate activities.
    a. Menu options:
        i. **Work at Task level**: switch from *activity level* to *task level*;
        ii. **Skip task**: see below;
        iii. **Close task**: when completed a *task* can be closed, prohibiting that users continue to work on *activities* of that *task* or start new *activities* in that *task*; only project *administrators* can re-open it;
    b. A panel that:
        i. Shows time management issues;
        ii. Allows to filter *activities* per domain;
        iii. Allows to filter *activities* per *user type* (role);
        iv. Allows to select an *activity* to start or redo; here several menu options allow you to
            - **Skip activity**: similar to skip task, see below;
            - **Finish activity**: when an activity is finished, a small window pops-up showing the time between start and end of the combined activities and allowing to fill-in time spent;
            - **Manage time**: a small window pops-up showing the time between start and end of the combined activities and allowing to fill-in time spent;

            Furthermore users work on (parts of) the activity by:
                - Providing *detail, actions and outcomes* of that *activity*;
                - Selecting *methods* used;
                - Attaching files to the *modelling journal*, accessible from the *task* at hand; the attached files can be retrieved, edited and uploaded again
                - In case of a *decision task* and *review task*, deciding to continue with the next *task* or go back and redo (a part of) the previous *tasks*
3. **Skip Task**:
    a. Menu options:
        i. **Perform task**: still perform task;
        ii. **Work at activity level**: switch from *task level* to *activity level*;
    b. Provide a reason why to skip the task

Browsing through the *tasks* should be done in the tree view, but – depending on if the prescribed order in the guidelines is enforced, *tasks* have to be performed in the right order.

The menu item *project* facilitates many functions, including:

- Convert local projects (on a stand-alone computer) to an online project;
- Export an online project to local;

- Save projects as project templates, which include all project characteristics (domains, tasks, etc., but no users);
- Access the scoreboard (project administrators can edit the default questions, managers can change the weights per step and per question and others can see the questions and results, if authorized);
- Change settings on whether or not confirmations are needed for all sorts of user interactions;
- Logon as a different user;
- Change password;
- Collapse the tree into tasks or expand it;
- Select small medium or large tree icons;
- Change project settings, which allows project administrators to adapt all project settings, which are specified during project set-up, e.g. to add new team members, edit authorizations, etc.

7.4.6 Reporting

During the project and after completion there is a strong need for fancy reports of the project. All reports will be generated from the content of a process (model) journal. Two aspects of a report are of interest here: its intended use and its audience. The use may vary from informing other team members, management checks on progress, to the final report for the client of the study done in the project. Audiences are also diversified, including scientists, managers, professional engineers, (lay) stakeholders and interested members of the general public.

Selecting what information has to be included in reports for specific use and audience is the major functionality of the report component. The selecting mechanism of the reporting component is very powerful, but also rather complex (Figure 7-8 and Figure 7-9).

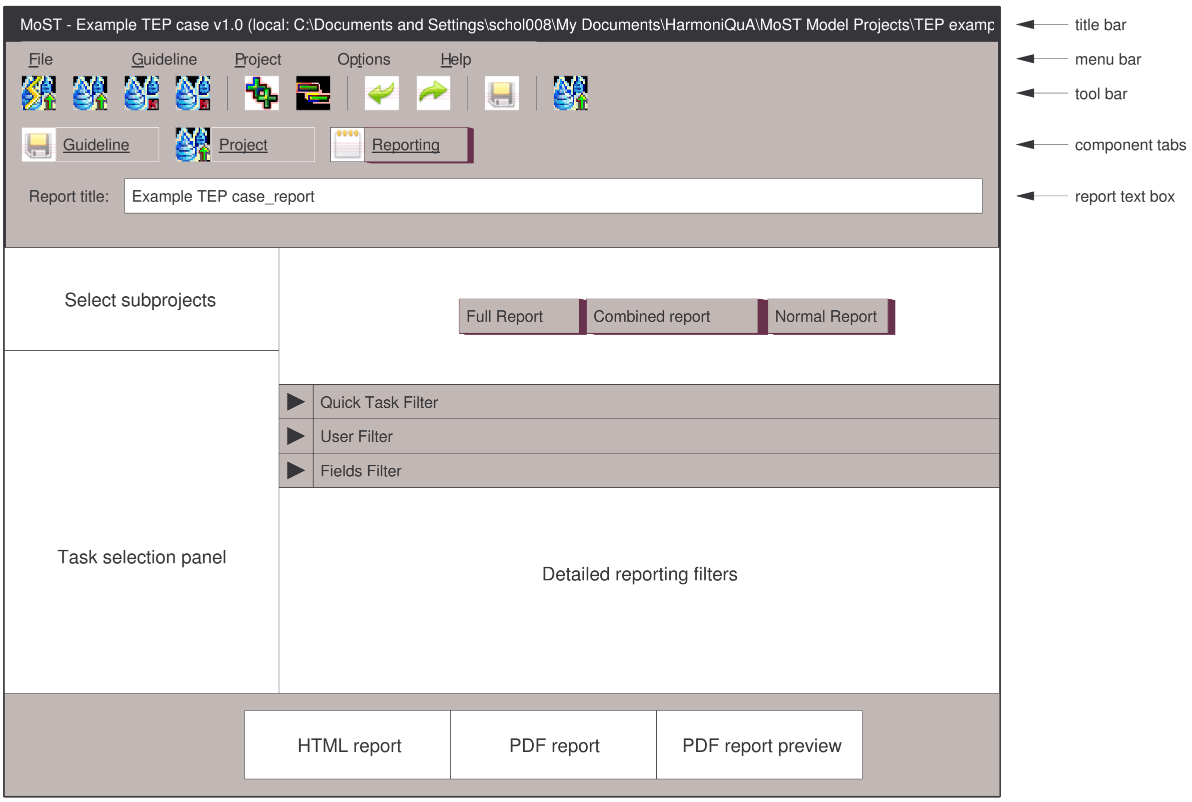


Figure 7-8. Schematized layout of MoST GUI of the report component in case a project (TEP case (GE)) has been opened with menu bar, tool bar, component tabs and report text box. Centrally selection options and in the bottom three options to generate reports.

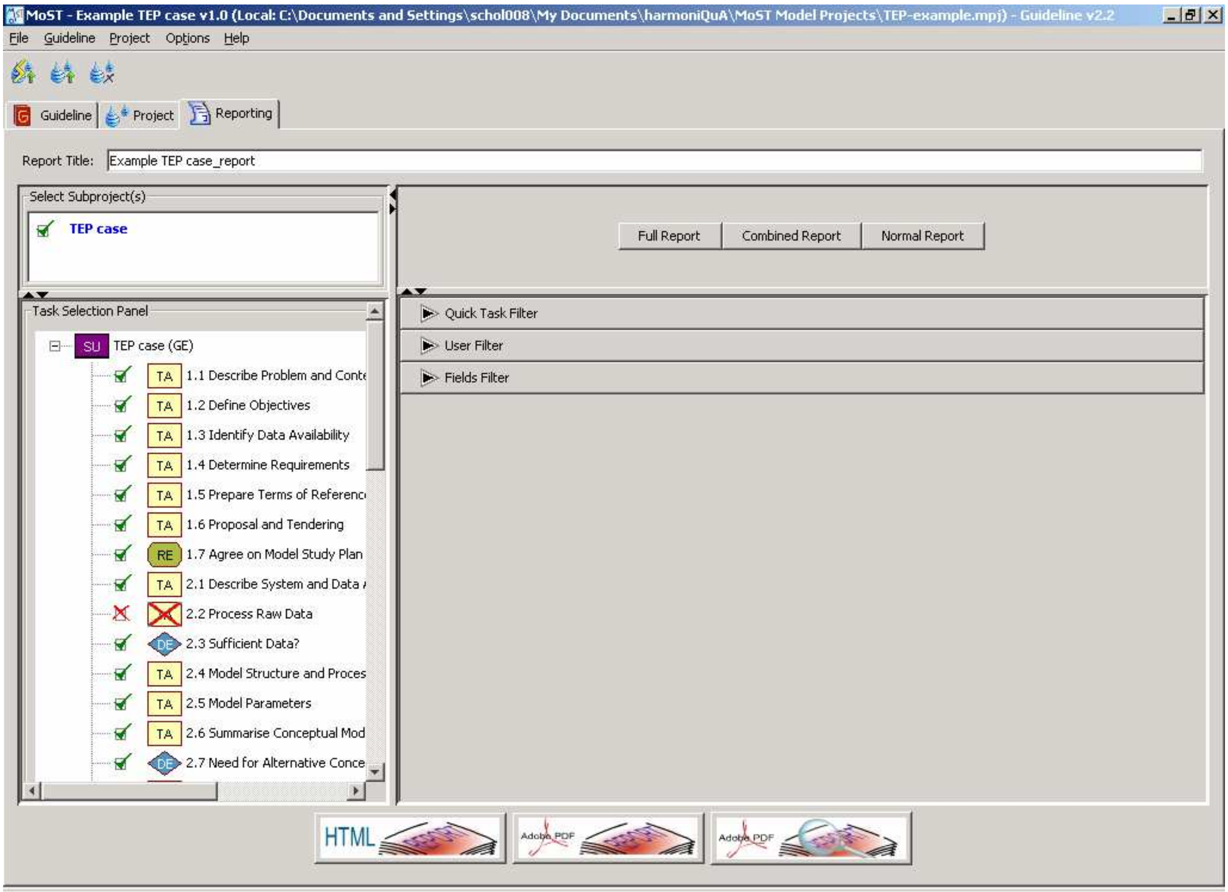


Figure 7-9. Screen dump of MoST GUI, see Figure 7-8 for explanation.

7.4.7 Additional functionality

In a later stage an advisory feature in MoST will derive advice from previous model studies, of which model journals are stored in a model archive, as extra help on how to perform the model study at hand. This feature is still in the design stage, but it will help modelling teams to learn from the past.

MoST is a complex tool and novice users need some training before they can benefit from all its features to support modelling teams in their daily routine. Therefore, comprehensive training material has been developed for students and professional modelers in water management[19]. This training material consists of written material, presentations, exercises that encourage using MoST in a test case project and many screen-recording movies on MoST, its KB, a case study and some background information. The screen recordings are Macromedia Flash® movies with a typical length of one to two minutes, usually accompanied by an instructor's voice. The movies allow trainees to work individually, at their own pace. Using the movies also allows the instructor both to explain in words and to show by demonstration. The movies are the core part of the training material and aim at helping users to work with MoST and will act as a sort of animated help facility.

In addition to the training material a help function has been implemented with the summarized information of the training material.

7.5 Ontological layers 5

In Chapter 6 an ontological framework for processes in which persons co-operate in multidisciplinary teams has been proposed. What such teams do is stored in modelling journals. The structure of such modelling journals (ontological layer 4) is based on ontological layer 3 (discussed in Chapter 6). There are no new concepts in ontological layers 4 and 5. The concepts introduced in ontological layer 3 (Chapter 6) get new (extra) properties that are used to store what multidisciplinary teams do in a project. These properties are

[19] The training material is accessible from www.HarmoniQuA.org/training.

discussed in section 7.4. Figure 7-10 shows a structure diagram of ontological layer 4 of the process ontology.

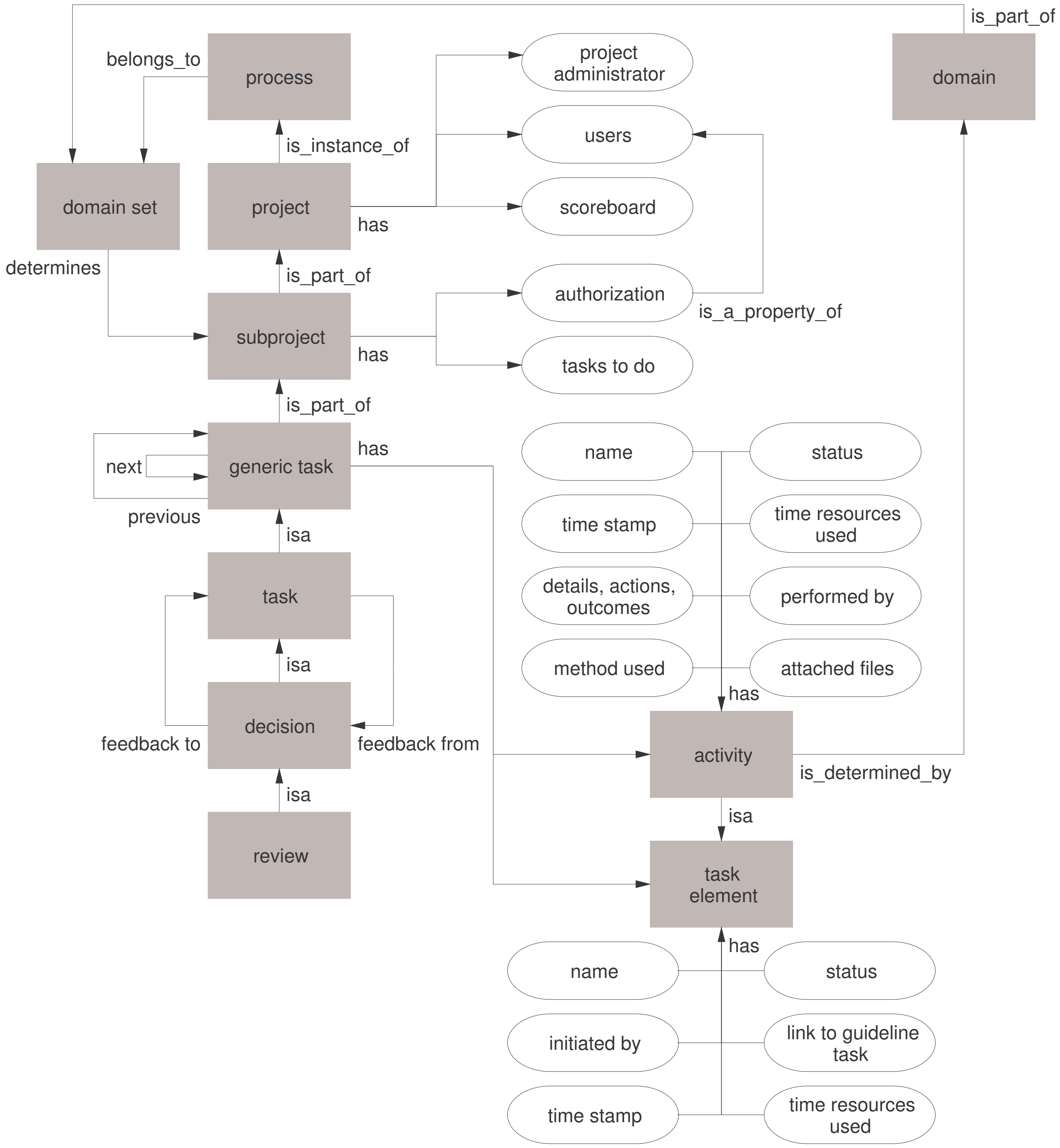


Figure 7-10. Structure diagram of the modelling journal ontology, using the meta-ontology of Chapter 5, *ontological layers 1*, *2* and *3* of the process ontology (instantiated for *simulation modelling*), discussed in Chapter 6. Rectangles are *concepts*, ovals are *properties* and arrows are *relations*. Shadowed *concepts* are detailed in *ontological layer 3* (discussed in Chapter 6). Some *concepts* and *properties* are left out.

7.6 The scope of the Modelling Support Tool

In this chapter the Modelling Support Tool, MoST, has been discussed. Together with the KB for modelling, discussed in the previous chapter, it forms a technology to support complex processes, in which teams have to cooperate. Describing the scope of MoST entails scope items of the KB too and therefore this section will outline the scope of the Modelling Support Tool with some detours to the process KB.

The technology, composed of the Modelling Support Tool, its KB and the KB-editor have been designed to collect, structure and present knowledge for model-based problem solving for water management and running modelling projects. It is not designed for other disciplines or application domains than water management and the types of models common in this field. Furthermore, the KB and Modelling Support Tool have only been tested on usefulness in this context, which is discussed in Chapter 10. However, the KB and Modelling Support Tool have been designed in a way that allows extending the KB for modelling to other application areas, but this has never been demonstrated or tested so far.

The Modelling Support Tool, MoST has been tested for model-based water management (see Chapter 10). To identify the scope of MoST, two questions have to be answered.

First: is MoST useful for other types of modelling than multidisciplinary model-based water management? At this stage this question cannot be answered, as it is not tested yet. Nevertheless its use for other types of models and modelling seems promising, although proof of such usability requires further testing. Without profound testing, the answer seems positive, as the Modelling Support Tool does hardly contain any water management characteristics and is rather generic. For example: the present tool has been tested in four university courses[20] not directly related to model-based water management.

The second question on the scope of the Modelling Support Tool can be summarized as follows. Can this tool (and the associated MoST-technology) also be used for other processes than modelling?
Processes to be supported by the MoST-technology resemble business type of processes, in which persons have to cooperate in order to complete some complex task. In this way the Modelling Support Tool would be used as a kind of workflow management system (see Chapter 2), but the type of application domain, for which it is interesting to test the MoST-technology (i.e. the KB structure and the Modelling Support Tool, but not the content of the KB), will be different. Interest has been shown for this technology by some *research technology and development projects*[21], e.g. AquaStress, but other types of application could possibly also be interesting, e.g. medical protocols, protocols for governmental or local authorities, which are complex, because they require large amounts of knowledge and knowledge related expertise. This does not mean that the MoST-technology aims at facilitating expert systems, as in the MoST-philosophy, users, i.e. human persons, with various roles including experts, professionals, decision makers, supervisors and controllers, are the key players and vital components of this kind of systems. In order to be used optimally in other cooperation processes, MoST has been extended in functionality and is at present available as Process Support Tool, ProST. This will be briefly discussed in Chapter 11.

Despite the interest shown for the Modelling Support Tool or its underlying technology, it's not an easy task to 'sell' the concept of ontological process support to others. Why should anyone use it? What's new? This type of questions can not be convincingly communicated with persons unfamiliar with the developed technology. To overcome these appreciation barriers, training material has been developed. The developers of the MoST-technology experienced that people, incidentally confronted with it, are very interested and want to learn more on its potential for their own 'business', the complex processes they are confronted with in every day's life.

7.7 References

Glinz, M., 2005, Rethinking the Notion of Non-Functional Requirements, Proceedings of the Third World Congress for Software Quality (3WCSQ 2005), Munich (D), 55-64, http://www.ifi.unizh.ch/groups/req/ftp/papers/3WCSQ2005.pdf.

Refsgaard, J.C. and H.J. Henriksen, 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.

Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.

[20] MoST has been used in the following courses at Wageningen University: INF-20306 *Programming and Modelling*, INF-31806 *Models for Forest and Nature Conservation*, GRS-30306 *Spatial Modelling* and INF-30806 *Advanced Modelling and Simulation*.

[21] RTD-project is the term used in EU Framework Programs. Research technology and development project includes research as is common in natural science, but also more design oriented research typical for computer science and engineering.

Scholten, H. and A.J. Udink ten Cate, 1995. Testing simulation modelling quality, Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium, pp. II.A.3.1-6.
Scholten, H. and A.J. Udink ten Cate, 1996. Quality assessment of simulation modelling. In: C. Lokhorst, A.J. Udink ten Cate, A.A. Dijkhuizen (Eds.), Information and Communication Technology applications in Agriculture: state of the art and future perspectives. ICCTA, Wageningen, pp. 353-358.
Scholten, H. and A.J. Udink ten Cate, 1999. Quality assessment of the simulation modeling process. Computers and Electronics in Agriculture 22, 199-208.
Scholten, H. and S.A. Osinga, 2001. How to support GMP in model-based DSS. In: M. Makowski (Ed.), Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, August 27-29, 2001. IIASA, Laxenburg, Austria, pp. 45-47.
Scholten, H. and S.A. Osinga, 2002. Requirement analysis report of the HarmoniQuA knowledge base and tools. Wageningen University, Wageningen, 102 pp.
Scholten, H., 1994, Quality assurance of the simulation modeling process, A.R. Kaylan, A. Lehmann, T.I. Ören, Proc. of the European Simulation Symposium, Istanbul, October 9-12, 1994, Istanbul, Vol. I, 85-89.
Scholten, H., 1999. Good Modelling Practice. In: M. Makowski (Ed.), Abstracts of the 13th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment. IIASA, Laxenburg, pp. 57-59.
Scholten, H., 2001. An ontological approach in quality assessment of modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering. TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0, Delft, The Netherlands.
Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.
Scholten, H., J.C. Refsgaard and A. Kassahun, 2004. Structuring multidisciplinary knowledge for model based water management: the HarmoniQuA approach. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Proceedings of iEMSs 2004: "Complexity and Integrated Resources Management". International Environmental Modelling and Software Society, Osnabrück, Germany, June 2004, pp. 1288-1293.
Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, H. Wösten, R.D. Koeze and J.J. Noort, 2000. Good Modelling Practice in water management, Proceedings HydroInformatics2000. International Association for Hydraulic Research, Cedar Rapids, Iowa, USA.
Scholten, H., R.H. Van Waveren, S. Groot, F. Van Geer, J.H.M. Wösten, R.D. Koeze and J.J. Noort, 2001. Improving the quality of model-based decision support: Good Modelling Practice in water management. In: A. Schumann, J. Xia, M. Marino, D. Rosbjerg (Eds.), Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the International Association of Hydrological Sciences, Maastricht, The Netherlands, July 18-27, 2001. IAHS, Maastricht, The Netherlands, pp. 223-230.
Scholten, H., A. Kassahun and J.C. Refsgaard, 2006. Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006, Research Publishing, Volume 3, 2231-2238.
Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J. Noort, 1999. Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice. STOWA/RWS-RIZA, Utrecht/Lelystad, 90-5773-056-1, 149 pp.

8 Towards a problem ontology instantiated for bivalve ecophysiology

I do not know what I may appear to the world,
but to myself I seem to have been only a boy playing on the sea-shore,
and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary,
whilst the great ocean of truth lay all undiscovered before me.
(Isaac Newton, 1643-1727)

Content of Chapter 8

This chapter is based on parts of the following papers:

Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.

Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.

Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.

Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (*Cerastoderma edule* L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.

8.1 Introduction

The ontology presented in this chapter aims to describe the combination of problem, object system (OS) and relevant domain knowledge, necessary to solve the problem. Such a problem ontology makes knowledge on problems and associated object systems explicit for two main purposes: reuse of (parts of) the knowledge and enabling or improving the communication between teams involved in problem solving. As part of the overall problem solving ontology (see Figure 8-1 and for more details Chapter 4, section 4.6 and Figure 4.6) the problem ontology uses the vocabulary of the meta-ontology (Chapter 5) and it should cooperate with the modelling ontology of Chapter 6 and the model ontology of Chapter 9. Just as in Chapter 6, ontologies consist of an ontological structure and of ontological content. The latter can be called a knowledge base (KB) as is customary in the field of knowledge systems and AI.

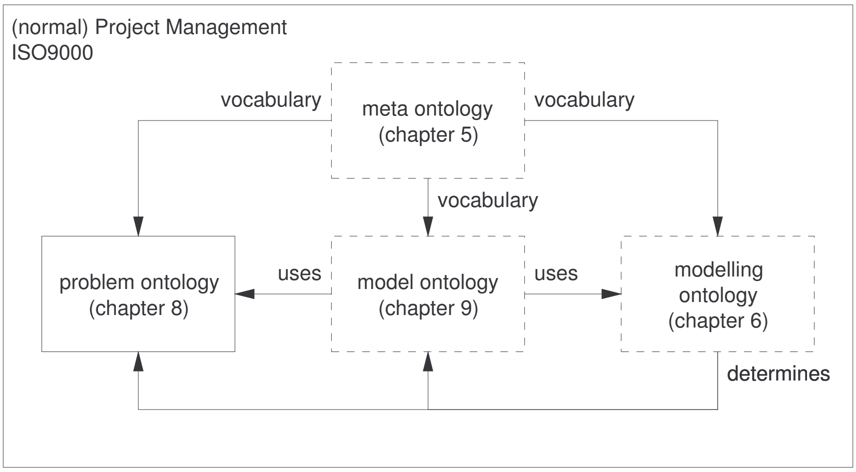


Figure 8-1. Linkage between the problem ontology of this chapter and the various other ontologies.

Problems are the main topic of the problem ontology, but this ontology is also about the real world, defined as a system, in which the problem is situated. This system will be named *object system*, as it is the object of the problem-solving process. Furthermore, this ontology contains *domain knowledge*, necessary to describe and solve the problem at hand.

Problem is defined in the meta-ontology (Chapter 5) as[1]:

> *An obstacle which makes it difficult to achieve a desired goal, objective or purpose. It refers to a situation, condition, or issue that is yet unresolved. In a broad sense, a problem exists when an actor becomes aware of a significant difference between what actually is and what is desired. Every problem asks for an answer or solution.*

Problems can be defined in two groups, those that need to be solved and those that aim at understanding the object system. The first group (problems that need to be solved) consists of practical problems: an unwanted situation has to be changed into a wanted one. Numerous examples can be given in various domains. A few will be listed here.

- In the *environmental domain*: algal blooms characterize an unwanted situation that first has to be understood and subsequently solved.
- If a *society* produces too much garbage it is clear that there are many ways to solve this problem into a wanted situation, i.e. 'not too much' garbage[2].
- In *companies* where a sales department cannot get along with production this unwanted situation should be changed in a wanted one by improving sales or adjusting production to sales, within technical production boundaries.
- Waiting queues for *hospital* treatment are also unwanted situations that have to be resolved.

[1] From Wikipedia: http://en.wikipedia.org/wiki/Problem.
[2] 'Not too much' should also be defined and made explicit and quantified.

- Files in *care organizations* are often incomplete, distributed and not up-to-date and therefore cause many problems, sometimes with high risks. Such an unwanted situation should not only be recognized and understood, but also solve, e.g. by implementing an Enterprise Resource Planning system.

The ontology proposed in this chapter aims at capturing and structuring knowledge about problems in their context[3]: the object system these problems belong to and its associated domain knowledge. The organization of this ontology should allow reuse of (parts of) it. By definition (Chapter 3), ontologies aim at formalizing and structuring knowledge and it is implicitly assumed that the structure and the group of users accept its content[4]. Unfortunately - in our case this group was very small - this ontology has been used in practice but has not been validated in a series of independent test cases. Its design is based on some case studies related to modelling studies in the past (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005). The expertise used in these studies has been collected in several other studies related to ecosystem modelling (Klepper, 1989, Scholten *et al.,* 1990, Van der Tol and Scholten, 1992, Klepper *et al.,* 1994, Scholten and Van der Tol, 1994) or studies using ecosystem modelling for environmental or shellfish culture related problems (Herman and Scholten, 1990, Van der Tol and Scholten, 1998).

This chapter will continue with a section on design requirements for a problem knowledge base. In section 8.2 an ontological structure for problems will be proposed. Section 8.3 - the main part of this chapter - will attempt to present a structure for a (reusable) ontology for problems. The first part after the introduction to the problem ontology describes the generic part of the problem ontology (section 8.3.2). The next section (8.3.3) specializes the generic problem ontology for problems to be solved by mathematical modelling. Section 8.3.4 focuses on problem knowledge for simulation models. In section 8.3.5 the ontological structure for bivalve ecophysiology will be instantiated. Section 8.3.6 describes some case studies for bivalve ecophysiology. Section 8.4 will evaluate the appropriateness of the proposed problem ontology.

The knowledge presented in this chapter is not new. It consists of commonly accepted scientific knowledge, which is generic or specific and published by many authors. The way of organizing this knowledge is the new element and facilitates its use and even enables reuse.

8.2 Design requirements for a problem knowledge base

The design of a problem knowledge base is a complex matter and should not lead to a single, monolithic knowledge base for problems, associated object systems and relevant domain knowledge. Organizing it in several layers of specialization facilitates communication between multidisciplinary model-based problem solving teams and enables reuse, especially of the more generic knowledge layers. Each layer provides language to describe concepts and relations in the next (more specialized) layer.

The structured ontological framework for problems should therefore be organized in ontological layers[5] (see Figure 8-2). More specialized layers are more generic and easier to reuse. The more specific higher layers are more difficult to reuse, but contain the more practical knowledge. The content is described in instances[6] of structural concepts[7]. Just as in the other ontologies (the *meta-ontology* of Chapter 5 and the *process ontology* instantiated for modelling in water management of Chapter 6), the more structural formal parts of an ontology have – by definition[8] – to be manageable for *machine actors* and *human actors* and the instances (i.e. the content of the KB) need only to be understandable for *human actors*.

The major design requirement comprises what concepts to include and which relations to allow between the concepts. Allocating the concepts in the right *ontological layers* is an extra requirement in our approach. In

[3] *Problem context* is defined in Appendix F.
[4] Tom Gruber in: *Lytras and Gruber, 2004*.
[5] *Ontological layer* is defined in the meta-ontology of Chapter 5.
[6] *Instance* is defined in the meta-ontology of Chapter 5.
[7] *Concept* is defined in the meta-ontology of Chapter 5.
[8] This definition is given in Chapter 3: *an ontology is a concise and precise, formal specification, shared by a group of persons and providing sufficient vocabulary that a piece of knowledge can be formalized for its purpose, is understandable for its human users and manageable for its machine users (computers).*

this way parts of the ontology can be reused in an easier way. Here the following *ontological layers* will be used:

- Ontological layer 0: *meta-ontology* with basic terminology;
- Ontological layer 1: *generic problem knowledge*;
- Ontological layer 2: *problem knowledge for mathematical models*;
- Ontological layer 3: *problem knowledge for simulation models*;
- Ontological layer 4: *problem knowledge for application domains*, i.e. instantiated with problem knowledge for bivalve ecophysiology;
- Ontological layer 5: *problem knowledge for projects*, i.e. instantiated with problem knowledge for bivalve ecophysiology projects.

Ontological layer 0 (meta-ontology) has been discussed in Chapter 5. *Ontological layer 1* (generic problem knowledge) and *ontological layer 2* (problem knowledge for mathematical models) *ontological layer 3* (problem knowledge for a specific application domain) are intended to be reusable. *Ontological layer 4* (problem knowledge for application domains) and *ontological layer 5* (problem knowledge for projects) are very specialized and useful only for specialists in that very specialized domain. Nevertheless these two layers are necessary to fill an example knowledge base to prove the usefulness of the whole ontological framework for problems and their associated object system and domain knowledge. Ontologies should be a shared body of knowledge usable for a group. Only its use can prove its merits.

Detailing *ontological layer 4* (problem knowledge for application domains) in *ontological layer 5* (problem knowledge for projects) requires very concrete and comprehensive scientific knowledge and should therefore be restricted to a specific piece of scientific knowledge, in this case: bivalve ecophysiology. Bivalves are those shellfish that have two shells. Some of these have commercial impact as they are eaten and represent therefore substantial commercial value. All have impact on coastal ecosystems where most species live. Bivalves play a dominant role in many estuarine and coastal waters, because of their high abundance and their huge impact on ecosystems (Gosling, 1992; Dame, 1996). Bivalves are used in many areas for water quality monitoring (Goldberg, 1975; Smaal and Widdows, 1994). Their commercial exploitation imposed questions on carrying capacity and ecosystem management (Korringa, 1956; Smaal *et al*., 1991; Héral, 1993), resulting in a large body of knowledge on the physiological ecology of bivalve filter feeders, to a large extent summarized in Smaal (1997).

Instantiating *ontological layer 4* of the problem KB with ecophysiological knowledge has been chosen for two reasons. Firstly, it is complex enough to be an adequate test case for the problem ontology. Secondly, developers of such ontology should have sufficient understanding of the domain itself. This cannot be acquired by reading only. As a result of several studies with experts[9] in the field I have (co)authored a series of papers and book chapters related to the ecophysiology of shellfish. Some of these publications deal with the role of bivalves in ecosystems (Klepper and Scholten, 1988, Scholten *et al.,* 1990, Van der Tol and Scholten, 1992, Klepper *et al.,* 1994, Scholten and Van der Tol, 1994 and others discuss the impact of ecological conditions on the physiology of bivalves (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005). Especially the latter papers on the ecophysiology of bivalves have functioned as a source of inspiration for the knowledge base on bivalve ecophysiology in this chapter. Most definitions of ontologies, given in Chapter 5, require that the knowledge in an ontology is shared by its users. Large parts of the knowledge in the problem ontology are derived from papers published in refereed journals and therefore these parts can be considered to be accepted by a substantial group of (potential) users, all experts in the field. This does not necessarily mean that the whole structured ontology, of which the content is this knowledge base on bivalve ecophysiology, is shared by all experts.

---

[9] The major expert on bivalve ecophysiology I had the privilege to cooperate with is Aad Smaal. Other experts to be mentioned are Olivier Klepper, Peter Herman, Marcel van der Tol, José Rueda, Theo Prins, Rob van Haren and many foreign experts, including Richard Dame and Mike Brylinsky, whose publications and discussions have strongly contributed to my knowledge on bivalve ecophysiology.

Unlike to the *project journals* (*ontological layer 5*) for modelling projects (Chapter 6 and Chapter 7), no similar ontological *pro forma*[10] (*ontological layer 5*) and associated tool will be developed for *ontological layer 5* of the problem ontology (problem knowledge for projects, i.e. instantiated with problem knowledge for bivalve ecophysiology in some simulation modelling projects[11]).

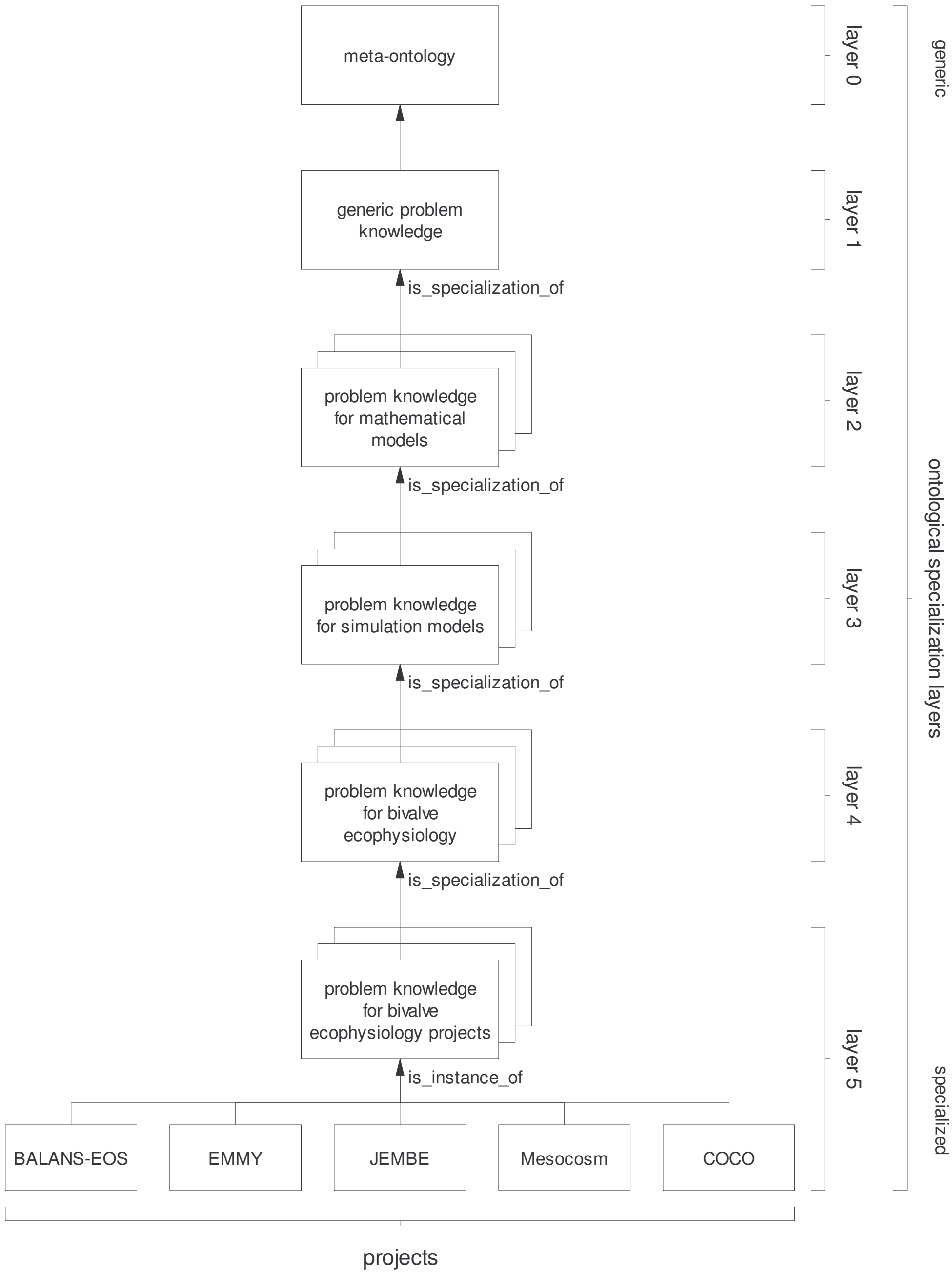


Figure 8-2. A stepwise ontology *specialization* with the problem ontology expanded to some more specialized ontological *concepts*. The top (meta-ontology) is the most *generic ontological layer* and the *concepts* at the bottom the most *specialized* ones. An *instance* is more *specific* than the *concept* where it is based on. This *structure diagram* is a simplification of Figure 4-6. Solid rectangles are concepts that will be described in detail; dotted rectangles are concepts and arrows *relations*.

[10] A 'pro forma' is a document template to be filled in for documenting and analysing options and impacts in a model study, standard practice for the professional modelling community in the UK. A *model journal* or a *project journal* can be seen as a *pro forma* of what has been done in a modelling project or a process oriented project in general. A problem pro forma can best be integrated with in a *project journal* in order to keep all relevant pieces together and available and make projects more transparent.

[11] Bivalve ecophysiology has been chosen to instantiate the problem ontology, a choice that is inspired by my own experience in this field of modelling.

8.3 A proposal for a problem ontology

8.3.1 An appropriate ontological framework

This section outlines the ontological structure for problems, the object system in which a problem is situated and the associated knowledge on the problem and the object system. The *ontological layers 1* (section 8.3.2) and *ontological layer 2* (section 8.3.3) and *ontological layer 3* (section 8.3.4) in this structure are the major parts and should be reusable to a large extent. *Ontological layer 4* (section 8.3.5) is focused on bivalve ecophysiology and only reusable in that context. *Ontological layer 5* (section 8.3.6) consists of a template for model-based problem solving projects and is therefore reusable. Its instantiations for practice oriented case studies on bivalve ecophysiology have only a recording character.

The framework with *ontological layers* surpasses the commonly adopted organization of ontologies in an ontological structure and instances of the concepts in the ontological structure, in which the content of the KB is located. In this way each *ontological layer* can be extended in an easier way or even replaced, making parts of the KB reusable for other problems.

The framework with ontological layers for the problem ontology is very explicit and facilitates reuse, but allocating concepts to an ontological layer appears to be somewhat arbitrary and dependent on the worldview and experience of the team accomplishing this task. Despite the arbitrariness, constructing criteria for the allocation of concepts to ontological layers is easy and straightforward (see Table 8-1 and also Figure 8-2). In *ontological layer*s *2*, *3*, *4* and *5* there are many alternative instances possible, but in each layer only one alternative is instantiated. Using *ontological layers* with increasing specialization to define problem knowledge allows to be generic in *ontological layers 0* and *1* and to go into very specialized details in *ontological layers 4* and *5*. Therefore the content of *ontological layer 4* and *ontological layer 5* are difficult to read for the non-specialists in bivalve ecophysiology. The value of this ontological framework with *ontological layer*s can only be demonstrated by filling it with real, scientific knowledge, which represents state-of-the-art knowledge with all available details.

In the next sections (8.3.2, 8.3.3, 8.3.4, 8.3.5 and 8.3.6) the five *ontological layers* will be presented. Each section consists of a *structure diagram* depicting the *concepts* and the *relations* between these *concepts* of that *ontological layer*. Definitions of all concepts are given in the tables of Appendix F.

Table 8-1. Allocation of concepts to the *ontological layers* of the problem ontology.

Ontological layer	Summary
0. Meta-ontology	• *Content*: terminology required for other ontologies • *Details*: Chapter 5
1. Generic problem knowledge	• *Content*: concepts, relevant for all problems and object systems (OS), e.g. OS entity, OS relation, OS aggregation, problem description, problem topic, problem context, etc. • *Details*: section 8.3.2
2. Problem knowledge for mathematical models	• *Content*: concepts relevant for (mathematical) models, e.g. static OS entity, variable OS entity, observable OS entity, physical knowledge, etc. • *Details*: section 8.3.3
3. Problem knowledge for simulation models	• *Content*: concepts related to physical processes, e.g. biological process (ecological process, physiological process, biochemical process), chemical process, transport process, etc. • *Instantiated for*: problems related to natural sciences, including environmental problems • *Alternative instantiations*: o Management science problem concepts; o Socio-economic problem concepts • *Details*: section 8.3.4

4. *Problem knowledge for application domains*	• *Content*: *concepts* specific for application domains, e.g. food related processes, respiration and excretion related processes, reproduction related processes, allocation and growth related processes.
	• *Instantiated for*: bivalve ecophysiology
	• *Alternative instantiations*:
	  o Aquatic ecosystems
	  o Air pollution issues
	  o Sustainable energy production
	• *Details*: section 8.3.5

5. *Problem knowledge for projects*	• *Content*: Extension of the knowledge of *ontological layer 4* for specific projects, i.e. a template for project characteristics.
	• *Instantiated for*: a series of projects related to the ecophysiology of mussels and cockles.
	• *Alternative instantiations*:
	  o Ecophysiology of oysters (*Crassostrea gigas*, *Ostrea edulis)*;
	  o Ecophysiology of cut trough shell (*Spisula subtruncata*).
	• *Details*: section 8.3.6

8.3.2 Ontological layer 1: generic problem knowledge

A number of *concepts*, related to the *problem ontology* have been introduced in the meta-ontology of Chapter 5. These include the following concepts: *problem*, *problem solving*, *problem owner*, *real-world problem*, *system*, *object system* (OS) and *role*. Many other *concepts* have to be defined, including those related to *object system* (*OS boundary*, *OS environment*, *OS context*, *OS aggregation level*, *OS structure*, *OS entity*, *OS relation*), to *problem* (*wanted solution*, *problem scenario*, *problem complexity*, *atomic problem*, *composite problem*, *problem description*, *problem subject*, *problem context*, *problem ownership*, *problem solving methodology*, *problem domain*) and related to *knowledge* (*domain knowledge*, *knowledge domain*, *functional knowledge*, *structural knowledge*, *process knowledge* (not to be confused with the modelling process; the *processes* meant here are physical *processes* in the OS).

The definitions of the *concepts* of *ontological layer 1* are given in Appendix F (Table F-1) and Figure 8-3 depicts the associated *structure diagram* of *ontological layer 1*. Together they represent the generic problem knowledge of the problem ontology.

8.3.3 Ontological layer 2: problem knowledge for mathematical models

Ontological layer 2 (problem knowledge for mathematical models) is more specialized than *ontological layer 1* (generic problem knowledge), but less specialized than *ontological layer 3*. *Ontological layer 2* focuses on concepts related to quantitative aspects of problems and object system. The definitions of the *concepts* of *ontological layer 2* are given in Appendix F (Table F-2) and Figure 8-4 depicts the associated *structure diagram* of *ontological layer 2*.

8.3.4 Ontological layer 3: problem knowledge for simulation models

Ontological layer 3 (problem knowledge for simulation models) is more specialized than *ontological layer 2* (problem knowledge for mathematical models), but less specialized than *ontological layer 4*. *Ontological layer 3* focuses on concepts related to processes relevant for the problem and object system at hand. The definitions of the *concepts* of *ontological layer 3* are given in Appendix F (Table F-3) and Figure 8-5 depicts the associated *structure diagram* of *ontological layer 2*.

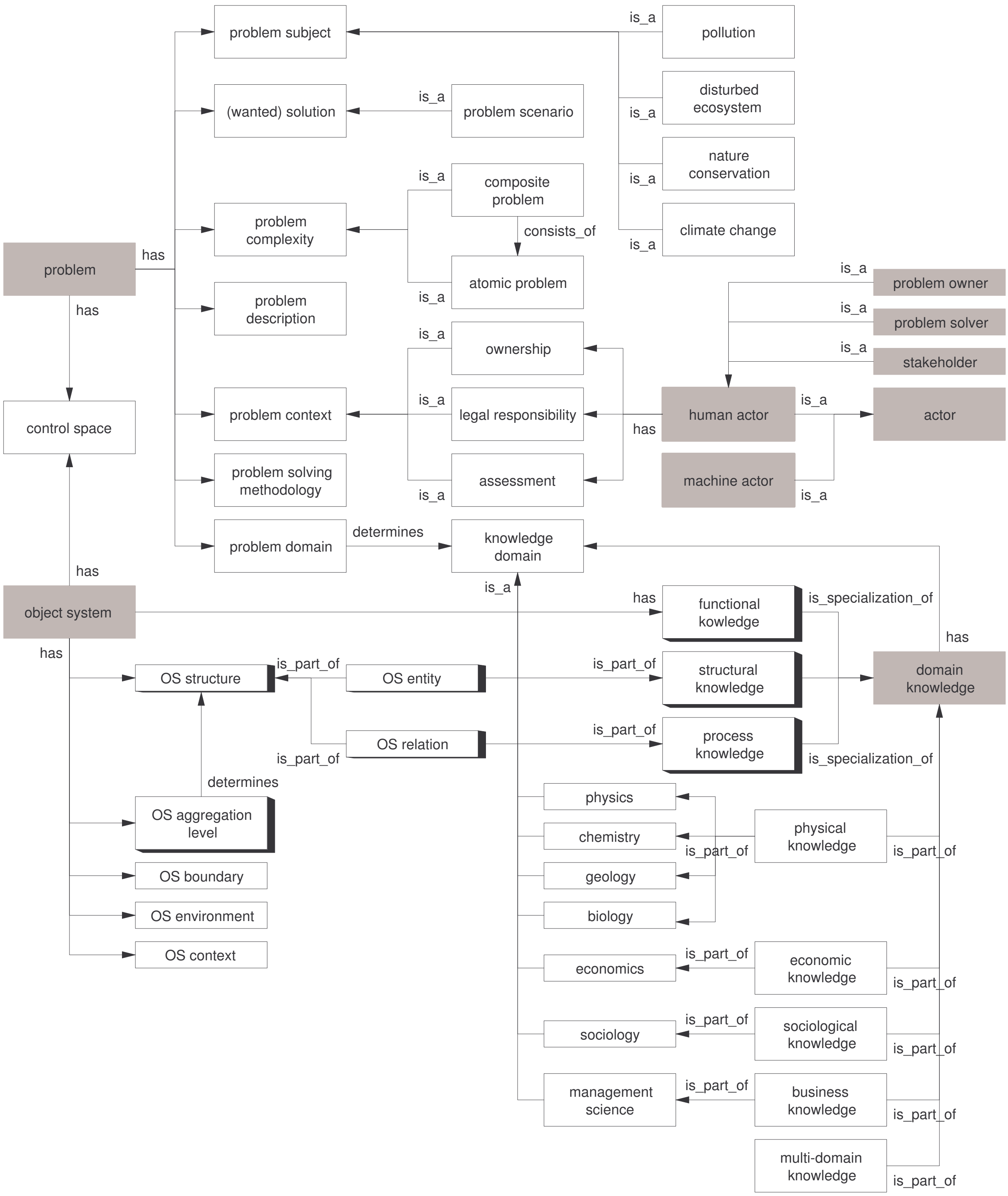


Figure 8-3. Detailed structure diagram of the generic problem ontology (*ontological layer 1*). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0* (meta-ontology of Chapter 5) and shadowed *concepts* are detailed in *ontological layer 2* and *3*. Some *properties* of *concepts* defined in Appendix F, Table F-1 are left out.

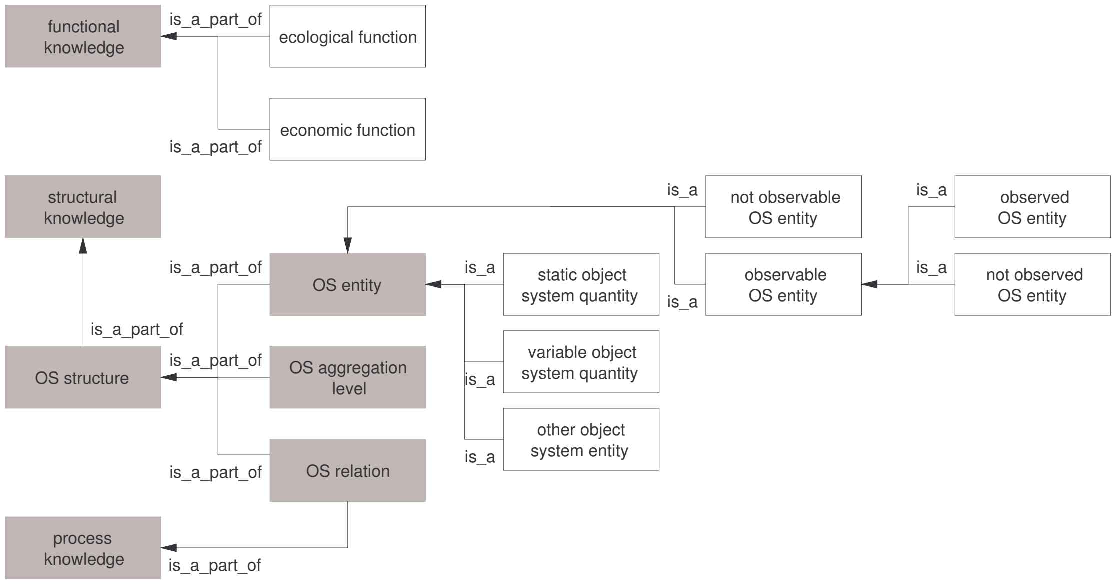


Figure 8-4. Detailed structure diagram of *ontological layer 2* (problem knowledge for mathematical models). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0* and *1*. Some properties of *concepts* defined in Appendix F, Table F-2 are left out.

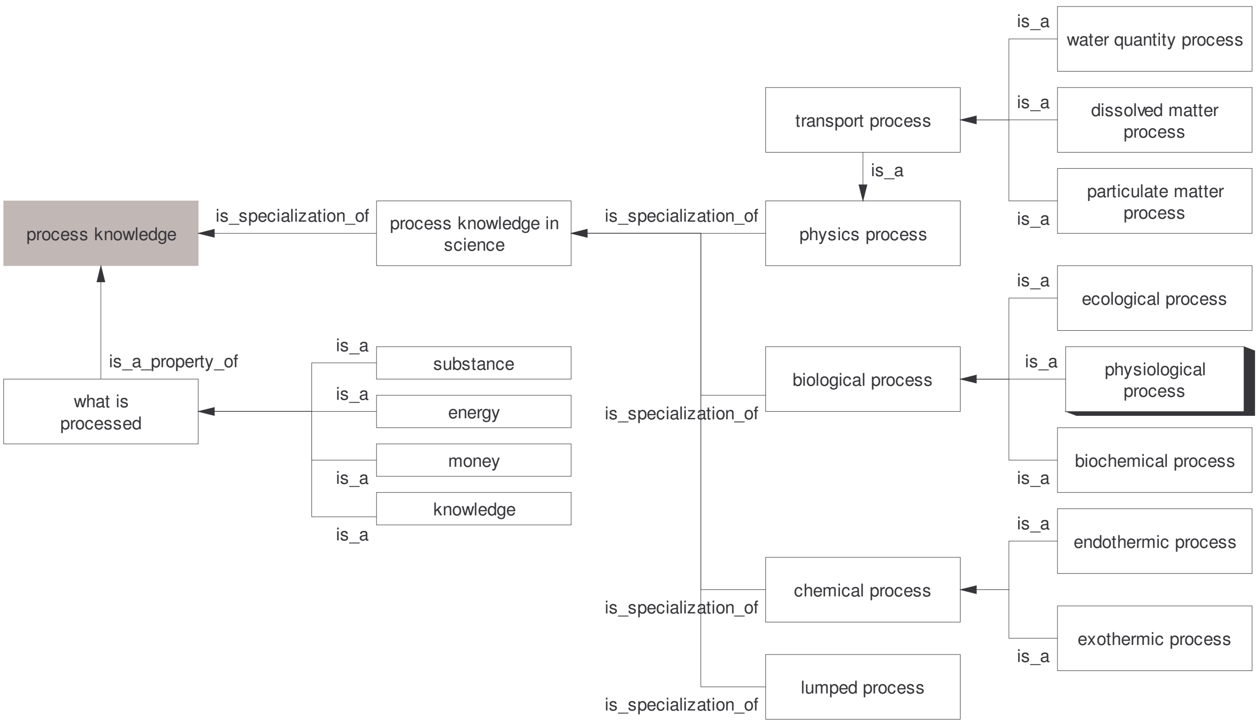


Figure 8-5. Detailed structure diagram of *ontological layer 3* (problem knowledge for simulation models). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0*, *1* and *2*. Shadowed *concepts* are detailed in *ontological layer 3*. Some properties of *concepts* defined in Appendix F, Table F-3 are left out.

8.3.5 Ontological layer 4: problem knowledge for application domains

8.3.5.1 Introduction

In specialized *ontological layer 4*, scientific knowledge on bivalve ecophysiology will be organized and described. To fit scientific knowledge into the structure of the problem ontology of especially *ontological layers 2* and *3* (sections 8.3.3 and 8.3.4), some scientific terms from biology have to be explained in this section.

Phylogenic taxonomy is the branch of biology dealing with lineages and evolutionary relationships of plant and animal species. It places species in a network of extinct and living other species and classifies these species in the taxonomic network. Bivalves can be classified in *Kingdom Animalia*, *Phylum Mollusca*, *Class Bivalvia*. The Class Bivalvia contains approximately 15,000 species[12].

Anatomy is the branch of biology dealing with the components of plants and animals. This can be done in different views. The most important views are looking at organs and looking at tissues. In ecology related modelling a more functional approach is often chosen with *functional parts* (related terms *functional compartment* and *functional group*).

Unlike components of many systems, bivalves have a clear dual function. The primary function is their role in an ecosystem, but many bivalve species have also an economic significance. This second role is often a reason to do research; at least the funding of research is easier in such cases.

8.3.5.2 Problem

As defined in the meta-ontology, problems are concepts that ask for a solution or understanding of the situation. Here understanding is the main goal, but summarizing scientific knowledge is a welcome side effect.

The problem belongs to the domain of eco-physiology of bivalve mollusks. These organisms have commercial value. Blue mussels and oysters are cultured and cockles are fished in relatively quiet parts of coastal areas in the Netherlands (Oosterschelde, Waddensea). These bivalves can deal with considerably varying ecological conditions. This flexibility and their commercial value make them an interesting topic for research. How can they cope with such extreme conditions? At present, the methods of fishing and culturing these bivalves are strongly criticized and these current methods will possibly be forbidden in the near future, as they are perceived as being endangering food supply of wading birds. Nature conservationists' interest will probably prevail over commercial interests of the bivalve industry. Alternative ways to secure the commercial interests in producing these species include artificial culture, as is common practice nowadays with many fish species and shrimps. Such an approach requires full understanding of the relations between ecological conditions and bivalve ecophysiology. Predicting bivalve growth, calculating the carrying capacity in a wide range of ecological conditions will facilitate artificial culture.

The *problem owner* is the mussel breeder and cockle fisher. They are also affected by the *problem* and therefore *stakeholders*, as their commercial businesses are endangered by governmental measures aiming at nature conservation, especially for wading birds as oystercatchers. There are a limited number of *domain experts* in the Netherlands mainly found at Wageningen Imares (consisting of the former WUR-RIVO and former WUR-Alterra), NIOZ and RUG. A part of the experts (mainly Wageningen Imares) is often seen as the scientific 'protectors' of the problem owner (both in finding scientific arguments to continue shellfish industry and for alternative production methods). The second group (mainly NIOZ and RUG) rather represents the nature conservationist's view on the problem.

---

[12] http://animaldiversity.ummz.umich.edu/site/accounts/information/Bivalvia.html, Animal Diversity Web. Accessed May 24, 2006.

8.3.5.3 Object System

Although the ultimate goal is to understand the relation between the ecological conditions and the growth of a whole population of bivalves, I will restrict the object system to a single bivalve organism, in order to simplify the problem by leaving out interactions between individual animals. They compete for food and substratum. They also struggle against each other to survive predation by birds, crabs, starfish and fish. Furthermore, they compete in their efforts to survive strong winter, storms and currents that endanger their attachment to the substratum.

The object system will be further discussed in the remainder of 8.3.5. All new concepts are defined in Appendix F (Table F-3) and the relations between the main concepts are depicted in section 8.3.5.7 (Figure 8-6).

8.3.5.4 Functional knowledge

8.3.5.4.1 Ecological function

Bivalves are not restricted to a single **habitat** type, as they are found in both fresh and salt water. Some, such as the oysters and marine mussels, have a reduced foot and live epifaunally or epibenthically, which means that they are permanently attached to a substratum; some, such as the clams and freshwater mussels, burrow slowly through the sand or mud using the foot; some, such as the cockle shells, live infaunally, buried in the upper layer of the sea bottom; while others, such as the shipworm, burrow through rocks or wood seeking protected dwellings and do damage to rock pilings and other marine installations. The scallops swim with great speed by suddenly clapping the shell valves together and ejecting water from the mantle cavity. Bivalves that are exposed at low tide, such as the marine mussels, keep their gills wet with water retained in the mantle cavity.

All bivalves have two **life stages**. In the first larval stage they live and feed in the water column and move as plankton[13]. After reaching the veliger stadium[14] they settle, often using byssus[15] threads to anchor.

The **environmental conditions** bivalve live in vary enormously. Here only the environmental conditions will be discussed, which blue mussels and cockle have to handle. They can survive a wide range of ecological conditions, varying significantly in temperature, food concentration, food quality and concentration of inorganic particles. Besides genetic adaptation[16] to these different conditions, the physiology of blue mussels has several mechanisms to deal with the environmental extremes. These mechanisms will be discussed in section 8.3.5.6.8.

Blue mussels and cockles are very abundant mollusk species, found in many coastal areas around the world. In Europe they flourish from Norway to the sub tropic Mediterranean. Mussels have to cope with a wide range of particles in the water. TPM, i.e. Total Particulate Matter (including food), can range from 1 g m^{-3} (Upper South Cove, Canada) to 350 g m^{-3} (Marennes-Oléron, France). The range in food concentrations is less than the range in TPM, varying between 0.3 g m^{-3} (Upper South Cove) and 7.5 g m^{-3} (Marennes-Oléron, France). Without doubt it is easy to find more extreme values.

Most natural blue mussels are attached to a hard substratum[17] or on top of sandy soils in subtidal or intertidal habitats, but if cultured, they live in dense beds on the slopes of the tidal channels (Netherlands) or are kept in various artificial constructions (France, Spain). Cockles live buried in the upper layer of intertidal or subtidal sandy soils or mud, buried until only the siphons project. Cockles and mussels can survive dry periods due to the tide. In the Netherlands cockles are most abundant on intertidal flats, which are dry during several hours of low tide. During the dry periods some processes stop, while other continue. This will be

---

[13] *Plankton* is defined in Appendix F, Table F-4.
[14] *Veliger stadium* is defined in Die.net (http://dictionary.die.net/veliger) as *any larval gastropod or bivalve mollusk in the state when it is furnished with one or two ciliated membranes for swimming*.
[15] Byssus threads are defined in Appendix F, Table F-4.
[16] Genetic adaptations are left out the proposed KB because it is irrelevant, as this KB is focused on the response of bivalves to varying ecological conditions.
[17] Only 5% of all mussels in the Oosterschelde live in natural habitats and 95% are cultured mussels.

discussed in 8.3.5.6. Mussels are normally subtidal which means that they live submerged and only a fraction of mussels in natural habitats will have to cope with dry periods.

Mussels and cockles (and to a less extent also other bivalves) have an enormous impact on coastal ecosystems (including estuaries) by filtering large amounts of particles from the water by pumping water through their gills. Mussels in the Oosterschelde pump at an average rate of 2 $l.h^{-1}.mussel^{-1}$. Their filtering activity can be enormous. At a biomass between 2-8 $gDW.m^{-3}$ bivalves can pump the total volume of their ecosystem in 2-4 days (Smaal, 1997). By filtering they remove substantial amounts of phytoplankton from the water. Herman and Scholten (1990) investigated the role of bivalves on estuarine ecosystems and they found that normally bivalves control phytoplankton biomass and prohibit algal blooms. Other phytoplankton eaters like zooplankton play a minor role following phytoplankton dynamics due to their shorter lifecycle. Under very eutrophic conditions phytoplankton can escape from the bivalve filtering control and may cause irreversible changes in the ecosystem.

Bivalves have considerable impact on the ecosystem they live in. They remove substantial amounts of organic material (dead or alive) from the water column. Partly they use that as food source and the main part is deposited on the bottom. What they eat is partly transformed in inorganic nutrients (N, P, Si) and CO_2. The parts deposited mineralize on the bottom, also releasing large amounts of inorganic nutrients with a positive effect on phytoplankton growth. In this way, bivalves play an important role in nutrient cycling. While bivalves reduce phytoplankton biomass, they also stimulate phytoplankton production (Smaal, 1997).

Bivalves also play a significant role in ecosystems as food source for birds, crabs, starfish and fish. Especially waders as the oystercatcher, *Haematopus ostralegus* (see Chapter 1) depend to a large extent to a diet of bivalves. In this way oystercatchers compete with human fishers of bivalves such as cockles in natural habitats.

8.3.5.4.2 Economic function

Many bivalve species are eaten, especially in coastal areas. Since historic times, they are collected and fished from their natural habitats. In the last 150 years also other practices are in use, first related to culturing oysters and later also culturing mussels. This culture is something between collecting and fishing from natural habitats and breeding in closed systems. Recent research is focused on organizing breeding commercially interesting shellfish completely separated from their natural habitats. These initiatives aim at a disconnection of a part of the shellfish culture from their natural habitats in order to facilitate conservation of these natural habitats and allow the natural predators (especially wading birds, e.g. oystercatcher) to get their share of this food source.

Shellfish are of commercial interest along all coasts, worldwide, but especially in Western Europe, and the both coasts of Northern America, although most intensive cultures are found in the Netherlands, France and Spain.

In the Oosterschelde (35,000 ha) mussels are cultured in mussel plots, on average of 1900 ha and standing stocks vary (average 4000 tons dry weight). The average yield of mussels in this area is 30,000 tons wet weight. A large part of the mussel culture (especially breeding of the younger stages) is located in the Dutch part of the Waddensea. In total, the mussel industry has a yearly production of 60 million tons fresh weight. A large part of these mussels is exported to Belgium and France[18]. Before 2005, cockles yield circa 30 million tons fresh weight (with large variations), almost completely exported to Spain and Portugal. At present there is no mechanical cockle fishery allowed in the Dutch part of the Waddensea. The yield of oysters is circa 2 million tons fresh weight, mainly Japanese oyster. In total, the bivalve industry had a turnover of € 266 million in 2003.

Since 2005, cockles are fished in the Oosterschelde, if more than 4 million tons are available, which is seldom the case. They are fished from the top layer of intertidal banks. Ships use a big fishing device, which removes the top layer and uses water to remove sand and other particulate material. In this way, cockle

[18] Total export circa M€ 200 per year in 2003.

fishers can fish their yearly ration in a few days. The Dutch government dictates this ration and aims at guaranteeing that ‘normal’ oystercatcher standing stocks will be able to feed on the leftovers.

8.3.5.5 Structural knowledge

8.3.5.5.1 Species

The taxonomy of bivalves can be summarized to Eukaryotes, Metazoa (i.e. animals), Lophotrochozoa, phylum Mollusca, class of Bivalvia. Bivalves are sometimes called Lamellibranchia. The name ‘bivalves’ refers to the two shells. They have no head, and are generally filter feeders. They have strong muscles for pulling their shells together when a predator threatens them.

‘Blue mussels’, *Mytilus edulis* L. and ‘cockles’, *Cerastoderma edule* (L.) are the most prominent examples of the more than 50,000 species of bivalves. Several other bivalve species may become of interest for this part on the problem ontology. These include the abundant ‘Pacific oyster[19]’, *Crassostrea gigas*, the nowadays rare ‘common European oyster’, *Ostrea edulis*, and the ‘cut trough shell’, *Spisula subtruncata*.

8.3.5.5.2 Anatomy

Bivalves consist of two matching calciferous half shells joined together at a hinge and held closed by a set of muscles. Like other mollusks, bivalves possess a hard exterior shell and no internal skeleton.

Within the shell is a fleshy layer of tissue called the mantle; there is a cavity (the mantle cavity) between the mantle and the body wall proper. The mantle secretes the layers of the shell, including the inner nacreous, or pearly, layer. Sometimes a pearl is formed as a reaction to irritation, by the depositing of nacreous layers around a foreign particle. The head is much reduced, without eyes or tentacles, and a muscular hatchet-shaped foot projects from the front end of the animal, between the valves. The foot is used for burrowing, and, in some bivalves (e.g., razor clams), to swim. Many bivalves have two tubes, or siphons, extending from the rear end: one (the incurrent siphon) for the intake of oxygenated water and food and one (the excurrent siphon) for the outflow of waste products. The two tubes may be joined in a single siphon, or ‘neck’.

The gills, suspended within a mantle cavity, are usually very large and function for food gathering (filter feeding) as well as for respiration. As water passes over the gills, tiny organic particles are strained out and are carried to the mouth.

Bivalves have a complete digestive tract; a reduced nervous system; a complete, open circulatory system with a chambered heart, arteries, veins, and blood sinuses; and excretory and reproductive organs. In most species the sexes are separate, and the eggs and sperm are shed into the water, where fertilization occurs. The larval stage is free-swimming and lacks a shell.

8.3.5.5.3 Functional compartments

Because the physiological knowledge on bivalves is limited and because ecological monitoring programs measure only a few standard variables of bivalves, the anatomic knowledge is restricted to a few components. These components are distinguished because of their functional role in bivalve physiology and are further referred to as **functional compartments**. Of these functional compartments two substances are relevant and interesting: carbon and nitrogen in the organic parts of a bivalve, as these are essential in any budget study of the role and the fate of food in bivalve ecophysiology. The use of functional compartments in ecological modelling g is not new, but introduced by Brylinsky and Sephton (1991) and is comparable with the DEB approach of Kooijman (1990)[20].

The **shell compartment** consists of two parts the *inorganic part* (i.e. the shell) and an *organic part* that produces the inorganic part. *Shell size* can be expressed in many ways, e.g. *organic shell weight*, *dry shell weight*, *total wet weight* (shell and organism’s biomass), *shell length* and *cooked flesh weight*. *Shell length*

[19] *Crassostrea gigas* (Pacific oyster) is also known as Japanese oyster.
[20] From Wikipedia: Dynamic Energy Budget (DEB) theory aims to identify simple quantitative rules for the organization of metabolism of individual organisms that can be understood from basic first principles. The word *Dynamic* refers to the life cycle perspective of the theory, where the budget changes dynamically over time.

and *total wet weight* are easiest to measure. The percentage *cooked flesh weight* of the *total wet weight* is an indicator for (commercial) bivalve quality.

Bivalves hardly contain fat. They use glycogen as storage to survive periods with low food supply. Therefore a storage compartment is defined, consisting solely of glycogen and referred to as the **storage compartment**. As glycogen is a carbohydrate that can easily be transformed to glucose, it has no proteins and does not contain nitrogen at all.

The reproductive organs consist of gonads (reproductive glands) that are sometimes filled with large amounts of reproductive cells (eggs and sperm, together called gametes). This functional compartment will be named **reproductive compartment**.

The rest of the body will be called **somatic compartment**, as it represents the whole body (i.e. soma), except *shell compartment*, *storage compartment* and *reproduction compartment*. The somatic compartment is therefore a remainder compartment containing all what does not fit in any of the other functional compartments, despite the complexity of its anatomy (see 8.3.5.5.2). All organs and tissues in this compartment can be treated in the same way, maybe except the stomach and intestines, usually together referred to as 'gut', as the gut plays an important role in some of the physiological processes (8.3.5.6).

8.3.5.6 Process knowledge

8.3.5.6.1 Introduction

The processes described in 8.3.5.6 summarize the most relevant physiological processes for bivalves. The selection of these processes is based – implicitly – on the use of pieces of knowledge in certain models (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005) and therefore mainly covering knowledge on blue mussels and cockles, but to a large extent it is also useful for other filter feeding bivalves. The knowledge originates from research and is just made explicit.

8.3.5.6.2 Allometric relations

The demand for food depends on metabolic requirements for **maintenance**, **growth** and **reproduction**, all mainly depending on bodyweight. These dependencies are no linear proportionalities but allometric relations, i.e. of the form $a.\text{size}^b$ with a and b coefficients of the allometric relationship[21]. Maintenance is expressed as *respiration* and *excretion* and these two processes have a b-coefficient of approximately 0.7. The clearance rate, the pumping of water through the gills, has a b-coefficient of approximately 0.5. The difference between these b-coefficients means that food intake of larger (probably older) does not fully cope with energy losses through maintenance, leaving less potential for growth and reproduction.

8.3.5.6.3 Food related processes

Bivalve suspension feeders collect their food by filtering and sorting particles (seston) from the water column. Seston is composed of inorganic silt, detritus (refractory and labile) and living algal cells (phytoplankton and maybe very small fractions of zooplankton). Between ecosystems (for bivalves most estuaries and coastal areas), the fraction of organic material (detritus and phytoplankton) in seston may vary enormously with relatively high organic fractions but sometimes with very small ones. In the latter case it is hard for bivalves to filter sufficient food from the seston. This fraction varies depending on the location, the season, storms and several other factors, making them difficult to predict (Smaal, 1997).

Bivalve **food** consists of the organic part of seston (*POM*, i.e. *Particulate Organic Matter*). POM consists of living *algal cells* (*phytoplankton*) and *detritus* (dead organic material). The latter has varying *food quality* for *bivalves*. The more recently deceased *algal cells* it contains, the higher the *food quality*. *Detritus*, which represents high food quality, can also easily be decomposed by organisms, including bivalves, *zooplankton* and bacteria) is called *labile detritus*. If the *decomposition* of *detritus* is more difficult and it has a lower *food quality*, it is called *refractory detritus*.

[21] In case the b-coefficient equals 1, the relationship is linear. This coefficient should be dimensionless.

Mussels pump water through their gills in order to clear it. The **clearance** rate is the amount of water per unit of time cleared from particles. Based on research several mechanisms are accountable for **clearance depression** (Prins *et al*., 1991). In cases of high seston concentrations and moderate food uptake during storms or ecosystem conditions clearance depression limits pumping of water through the gills. The colony forming flagellate phytoplankton species, *Phaeocystis spss*, also has a negative effect on *clearance rate*. The single algal cells are tightly connected in the colony and bivalves cannot process this food source. It is therefore energetically wise to reduce pumping in order to deal with this phenomenon. Clearance depression has also observed in case seston consists mainly or completely of organic material and no (inorganic) silt. This is only observed under artificial conditions in experiments. In that case the composition of food consists mainly or completely of phytoplankton (alive algal cells). The process of retaining particles by the gills is called **filtration**. Some of the filtered material goes into the gut (stomach and intestines) and this process is called **ingestion**. By selecting better food particles from the filtered material, mussels apply **pre-ingestive food selection** with some *selection efficiency*. The size of the gut (*gut content*, allometrically depending on body size) determines also what can be ingested. *Food* of high *food quality* in a sufficient amount will remain shorter in the *gut* than *food* with a lower *food quality*. What is not ingested of the filtered particles is enveloped in **mucus** and ejected as **pseudo-faeces**. What is ingested stays for some time in the *gut*, slowly moving in the direction of the *anus* and leaving the body as faeces. The time it is in the gut is often called *gut passage time*. In the stomach the ingested material is digested and in the intestine part of the gut basic food elements (simple carbohydrates and amino acids) pass the membranes of the gut wall. The process is called **absorption** and is the essential part of the feeding process. What is absorbed consists almost completely of organic material. The *absorption efficiency* is a rather complex issue. Willows (1992) assumes a complex mechanism for optimization of food absorption and maximizing net energy gain in bivalve filter feeders. Such a mechanism is not required to describe absorption efficiency properly. It is sufficient to assume that absorption efficiency depends (exponentially) on *gut passage time*. Material that was not absorbed leaves the body through the anus as *faeces*. This **faeces production** can be measured. Absorbed material can be used for various processes related to maintenance. What is not needed for maintenance can be used for **growth** or (depending of the season) for **reproduction**. Growth, therefore, occurs only if food intake allows it (see 8.3.5.6.6).

8.3.5.6.4 Respiration and excretion

Respiration consists of several subprocesses, including the gas exchange at the gills. Oxygen passes the gill membrane and transported by the blood to the cells, where it is used to oxidize carbohydrates. In the latter complex chemical reaction, energy is produced to supply for maintenance. Carbon dioxide, a rest product of the oxidation, is released to the water by passing the gut membrane.

The subprocesses outlined above are too detailed and irrelevant for the purpose of the problem ontology, specialized for problem knowledge on bivalve ecophysiology. For this purpose it is sufficient to distinguish between two forms of respiration. The first will be called **basal respiration** (or better called **rest respiration**; this part of the respiration depends allometrically of body size and exponentially of water temperature) and **routine respiration** (related to the activity of the organism letting it depend on *ingestion rate* of organic material). Finally, at some stage we assumed that bivalve organisms need extra energy during *spawning*. There seemed however no clear reason for this, as the rest respiration and the routine respiration could explain all variation experimentally found in respiration rates.

Excretion is the process to remove metabolites, mainly protein metabolism related, such as ammonium[22]. Excretion is observed in all species, but excretion of ammonium by subtidal blue mussels is a important process, as it affects other ecosystem processes, which is not true for cockles and blue mussels with intertidal habitats.

8.3.5.6.5 Reproduction

Bivalve **reproduction** generally centers on external fertilization following the release of sperm and eggs into the water column. New bivalves develop and feed for several weeks in the water column, before reaching what is referred to as the veliger stage, at which point they settle. Because of their size, settlement location is dictated in large part by water movement patterns. Many settling veliger larvae, especially in burrowing

[22] The role of ammonium here can be compared with that of urea in mammals.

bivalves, utilize byssal thread attachment for anchoring prior to adulthood and some maintain this characteristic into adulthood.

Two subprocesses can be distinguished. The first is called **gametogenesis** (the development of reproductive cells, gametes, consisting of egg cells and sperm cells) and the second **spawning** (i.e. the release of gametes). For simplification seasonal changes in the gonads, i.e. the reproductive organs, is included in *gametogenesis*, as they follow the same seasonal cycle.

Gametogenesis starts at the moment when the *storage compartment* exceeds a certain bodyweight fraction. Gametogenesis continues until *glycogen* in the *storage compartment* drops below a certain minimum fraction. Reproductive cells are released in spawning, when the water temperature is above about 10° Celsius.

8.3.5.6.6 Shell growth

The two shells of bivalves consist of a (large) inorganic part and a (small) organic part. The latter is responsible for the growth of the first. The organic part of the shell grows, if the absorption is sufficient for respiration and mucus-production. The surplus will partly be used for growth of the organic shell part. The growth of inorganic part of the shell is tightly coupled to that of the organic part and will therefore grow accordingly. Shell length can best be calculated from the weight of the (inorganic) shell.

8.3.5.6.7 Allocation and growth

Given the concept of *functional compartments* (discussed in 8.3.5.5.3) and the *processes* (discussed in 8.3.5.6.3, 8.3.5.6.4 and 8.3.5.6.5) the allocation of absorbed food can now be described.

Partitioning of absorbed food is organized as follows: first a part of *absorbed carbon* is used for *respiration* and *mucus production* and a part of nitrogen for *excretion*. If there is a shortage of carbon the shortage is reallocated from the *storage compartment* or even (in case of starvation) by resorption of structural body tissues, i.e. from the *somatic compartment*. The latter will be the case if nitrogen in the *absorbed food* cannot cover the demand by the *excretion*. If there is a surplus, the mussel can grow and a part is allocated to the *storage compartment*, which increases with age. The remainder is divided (in a constant proportion) to the organic part of the shell and to the structural body tissue.

In summary, the first needs are maintenance (*respiration* and *excretion*). If available food covers these first needs, any surplus is divided between the *storage compartment* and the *somatic compartment*. Shortages are balanced by using glycogen from the *storage compartment*, or, of *storage compartment* cannot cover by using parts of the *somatic compartment*. The somatic compartment will cover any shortage of nitrogen.

8.3.5.6.8 Mechanisms to deal with varying ecological conditions

The physiological plasticity of bivalves is remarkable as they can cope with a wide range of ecological conditions. Mussels can survive a wide range of ecological conditions, varying significantly in temperature, food concentration, food quality and concentration of inorganic particles. Besides (slow) genetic adaptation to these different conditions, the physiology of mussels has several mechanisms to deal with the environmental extremes (Scholten and Smaal, 1999). Eight short-term adaptation mechanisms can be distinguished. Three of these mechanisms depress clearance rate, at high concentrations of (1) inorganic suspended particles, (2) in the presence of significant numbers of Phaeocystis colonies and (3) in case of an almost pure phytoplankton diet. The first mechanism prevents a too high uptake of particles with low nutritional value, while the second and third mechanisms allow mussels to deal with extreme conditions. Pre-ingestive selection (4) enables mussels to enrich moderate food quality to higher organic content of the ingested food fraction. A flexible Gut Passage Time (5), i.e. short GPT for high quality food and long GPT for poor quality food or in case of too little food, averages organic particle removal in the gut. The absorption efficiency depends on this gut passage time: the longer *gut passage time,* the higher *absorption efficiency*. In this way it is an additional help (6) for bivalves to respond to varying food concentrations by smoothing the *absorption rate*. Two extra mechanisms enable mussels to endure poor food conditions. The rest respiration is assumed to be relatively low (7) compared to the routine respiration, which depends on absorption rate itself. Finally (8) the physiological processes related to food uptake and energy investment are assumed to be slightly influenced by water temperature. All of these mechanisms regulate energy absorption to an almost constant average level and allow bivalves to deal with extreme environmental factors.

8.3.5.7 A structure diagram of the problem knowledge for modelling bivalve ecophysiology

All concepts of the *ontological layer 4* were discussed in section 8.3.5. These concepts are defined in Appendix F (Table F-3) and the relations between these concepts are depicted in Figure 8-6.

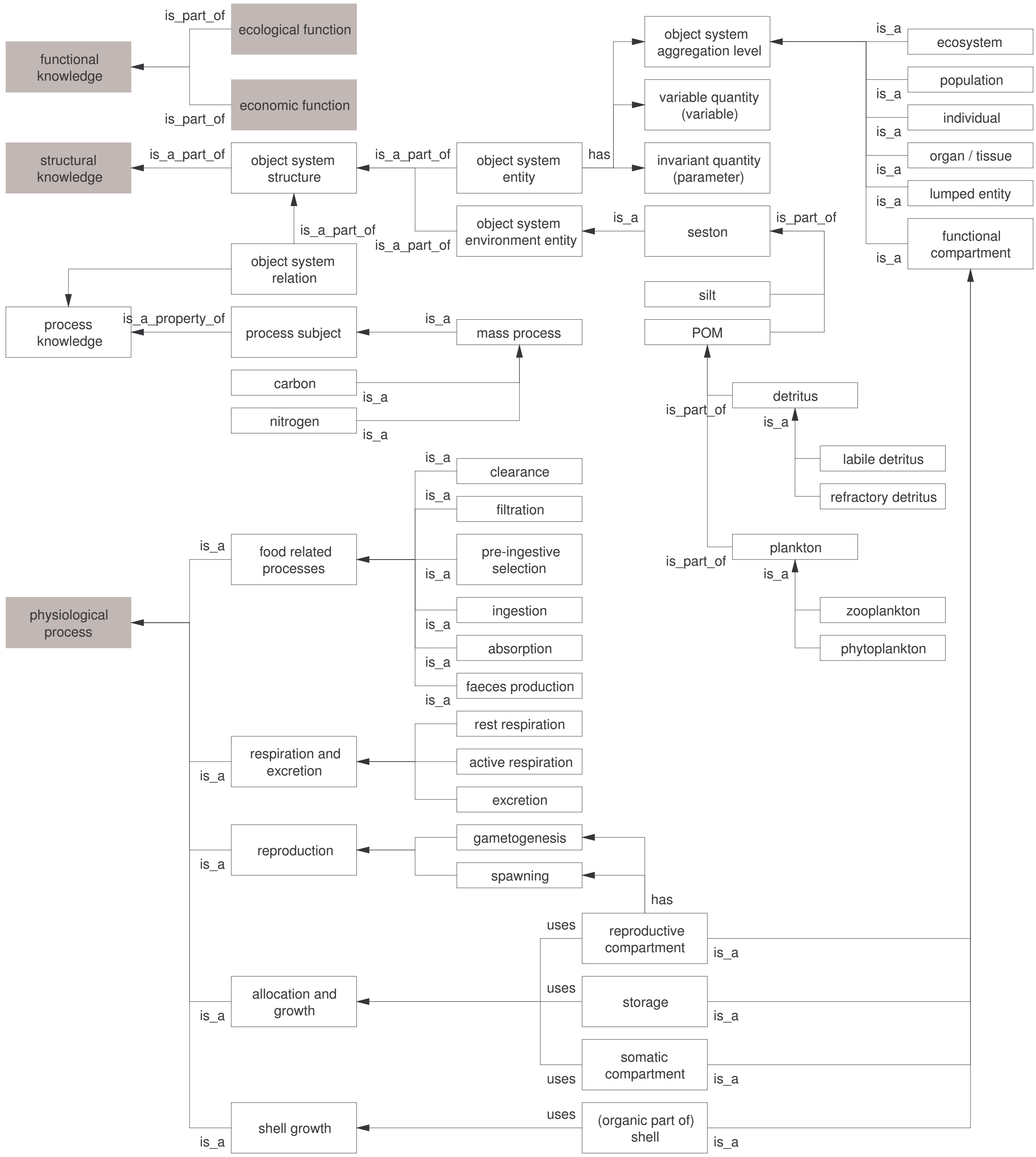


Figure 8-6. Detailed *structure diagram* of *ontological layer 4* (problem knowledge for application domains, instantiated with problem knowledge for bivalve ecophysiology). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layers 0*, *1*, *2* and *3*. Many concepts defined in Appendix F (Table F-4) and many relations are left out.

8.3.6 Ontological layer 5: problem knowledge for projects

8.3.6.1 Structure

The knowledge of *ontological layer 4* has been used in several scientific projects, all using a simulation model to understand a biological system, related to bivalve ecophysiology. Ontological layer 5 consists of a structural part in this section and instantiations of this structural part in section 8.3.6.2. The concepts presented in this structural part are rather generic, as they act as a template for projects on bivalve ecophysiology and ecological processes. Appendix F (Table F-5) gives an overview of the concepts in this project template. All concepts are self-explaining characteristics of (scientific) projects and will therefore not be discussed here.

8.3.6.2 Instantiations for bivalve ecophysiology projects

Instantiating the concepts of *ontological layer 5* results in a series of *instances*, being real projects on bivalve ecophysiology of which the information is filled in the template that represent the structure of *ontological layer 5*.

In the first project two bivalve species, the blue mussel *Mytilus edulis L.* and the cockle *Cerastoderma edule (L.)* are entities in an ecosystem (*object system*) with a substantial effect on that ecosystem. A part of the ecophysiological knowledge of bivalves as described in section 8.3.5 has been gathered in a large ecosystem study in the Oosterschelde, SW Netherlands (Klepper, 1989, Herman and Scholten, 1990, Scholten *et al.,* 1990, Van der Tol and Scholten, 1992, Klepper *et al.,* 1991, 1994, Scholten and Van der Tol, 1994, Van der Tol and Scholten, 1998). As one of the results of this project, an ecosystem model SMOES has been developed, which will also act as example model in Chapter 9. Details of this project are presented in Appendix F (Table F-6).

The ecophysiological knowledge of section 8.3.5 has been used for four case studies, three related to mussels and one to cockles. All four are based on the ecophysiological knowledge on bivalves as has been used in the ecosystem model SMOES and extended in order to include all available knowledge in a state-of-the-art model.

The first project relating the physiology of mussels to ecological conditions aimed at developing a simulation model for the growth of a single mussel under natural conditions, as observed in the Oosterschelde ecosystem, SW Netherlands (Smaal, 1997, Smaal and Scholten, 1997). The project characteristics are described in Appendix F (Table F-7).

In the second project the same knowledge on ecophysiology of mussels is tested in three object systems differing completely in ecological conditions. The results of this project have been published in a peer reviewed journal (Scholten and Smaal, 1998). The project characteristics are described in Appendix F (Table F-8).

In the third project on mussel ecophysiology the lessons learned from the previous studies have been tested in (semi) controlled mesocosm[23] experiments, leading to a final adaptation of the knowledge on bivalve ecophysiology, as described in section 8.3.5 (Scholten and Smaal, 1999). The project characteristics are described in Appendix F (Table F-9).

Subsequently, this body of (structured) knowledge as described in section 8.3.5 has been adapted for cockle, another bivalve species, and applied in a project. Cockles differ from mussels in terms of the bivalve KB (section 8.3.5) in their habitat, at least in Oosterschelde conditions, where mussels are always submerged, allowing them pump water along their gills continuously, while cockles live here mainly infaunally, buried in the upper layer of intertidal flats, which fall dry during longer or shorter periods at low tide, depending on

[23] http://www.seagrant.sunysb.edu/BTRI/btriterms.htm#m: a 'mesocosms' is an experimental apparatus or enclosure designed to approximate natural conditions, and in which environmental factors can be manipulated. A description of the mesocosms used for this project can be found at http://www.nioo.knaw.nl/cemo/phase/mesocosm/mesophase.htm.

flat elevation. On Oosterschelde tidal flats, the cockle habitat in the present research study, cockles are submersed for 16.8 $h.d^{-1}$. The project characteristics are described in Appendix F (Table F-10).

8.4 Scope and appropriateness of the proposed problem ontology

As for all knowledge bases it is hard to determine the scope and the appropriateness of the proposed problem ontology. The setup in ontological layers (Figure 8-2) allows discussion of the scope of each ontological separately. This approach makes problem and object system oriented knowledge at least explicit.

The meta-ontology (*ontological layer 0*) has already proven its usefulness in the process ontology (Chapter 6). For the problem KB in this chapter it is useful too, which indicates its appropriateness as set of basic terminology to bootstrap the other ontologies (process, problem, model). *Ontological layer 5* structures and instantiates real projects, which proves its appropriateness. That leaves *ontological layers 1*, *2*, *3* and *4* to be assessed.

Distributing concepts over these four *ontological layers* is somehow arbitrary. In *ontological layer 4* all real scientific application domain related knowledge has been organized, here knowledge on bivalve eco physiology. This problem knowledge for application domains consists mainly of instances of the structural concepts defined in *ontological layer 3* or in *ontological layer 2*. Concepts are allocated to *ontological layer 2*, if they are not generic but problem and object system aspects that can be represented in models, except for (physical) process knowledge, to be represented by simulation models. These concepts are allocated to *ontological layer 3*. Reuse of concepts in the more generic *ontological layers 1, 2 and 3* can be expected to be easier than the more specialized *ontological layers 4* and *5*.

The content of *ontological layer 4* will be tested by using its content as basis for the development of models on mussel and cockle ecophysiology in Chapter 9. A further test can be reusing *ontological layer 4* in the modelling of other bivalves, for which no models have been developed so far. Most interesting candidates are oysters (*Crassostrea gigas*, *Ostrea edulis),* the 'cut trough shell', *Spisula subtruncata* and *Ensis directus.* These bivalves are of some economic interest and are also an alternative food source for waders and ducks

Testing the usability of *ontological layers* 1 and *2* of the problem ontology can best be done by using these to build a knowledge base for completely different problems.

8.5 References

Brylinsky, M. and T.W. Sephton, 1991. Development of a computer simulation model of a cultured blue mussel (Mytilus edulis) population. Science Branch, Department of Fisheries and Oceans, Gulf Fisheries Centre, Moncton, New Brunswick, 81 pp.

Dame, R.F., 1996. Ecology of marine bivalves: an ecosystem approach. CRC Press Inc., Boca Raton, FL, USA.

Goldberg, E.D., 1975. The mussel watch - a first step in global marine monitoring. Mar. Poll. Bull. 6, 111.

Gosling, E. (Ed.) 1992. The mussel Mytilus: ecology, physiology, genetics and culture. Elsevier, Amsterdam, 589 pp.

Héral, M., 1993. Why carrying capacity models are useful tools for management of bivalve molluscs culture. In: R.F. Dame (Ed.), Bivalve filter feeders in estuarine and coastal ecosystem processes. NATO ASI series G, pp. 455-478.

Herman, P.M.J., Scholten, H., 1990. Can suspension-feeders stabilise estuarine ecosystems? In: Barnes, M., Gibson, R.N. (Eds.), Trophic relations in the marine environment. Aberdeen University Press, Aberdeen, pp. 104-116.

Klepper, O. and H. Scholten, 1988. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (SW Netherlands), DGW/DIHO, Middelburg, 278 pp.

Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary, DGW-Ministry of Transport and Public Works. Wageningen University, Wageningen, 270 pp.

Klepper, O., H. Scholten and J.P.G. Van de Kamer, 1991. Prediction uncertainty in an ecological model of the Oosterschelde estuary, S.W. Netherlands. Journal of Forecasting 10, 191-209.

Klepper, O., Van der Tol, M.W.M., Scholten, H., Herman, P.M.J., 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.
Kooijman, S.A.L.M. and R.J.F. Van Haren, 1990. Animal energy budgets affect the kinetics of xenobiotics. Chemosphere 21, 681-693.
Korringa, 1956. Oyster culture and biological productivity. Rapp. P-V. Réun. Cons. perm. int. Explor. Mer 140, 30-31.
Lytras, M.D. and T. Gruber, 2004. Every ontology is a treaty - a social agreement - among people with some common motive in sharing. Official Quarterly Bulletin of AIS Special Interest Group on Semantic Web and Information Systems 1, 1-5. http://tomgruber.org/writing/sigsemis-2004.pdf.
Prins, T.C., A.C. Smaal and A.J. Pouwer, 1991. Selective ingestion of phytoplankton by the bivalves Mytilus edulis L. and Cerastoderma edule (L.). Hydrobiol. Bull. 25, 93-100.
Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (Cerastoderma edule L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.
Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.
Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.
Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.
Scholten, H., O. Klepper, P.H. Nienhuis and M. Knoester, 1990. Oosterschelde estuary (S.W. Netherlands): a self-sustaining ecosystem? In: D.S. McLusky, V.N. De Jonge (Eds.), North Sea-estuaries interactions. Kluwer Academics Publishers, Dordrecht, pp. 201-215.
Smaal, A.C., 1997. Food supply and demand of bivalve suspension feeders in a tidal system, Wiskunde en Natuurwetenschappen. Rijksuniversiteit Groningen, Groningen, 237 pp.
Smaal, A.C., M. Knoester, P.H. Nienhuis and P.M. Meire, 1991. Changes in the Oosterschelde ecosystem induced by the Delta works. In: M. Elliott, J.-P. Ducrotoy (Eds.), Estuaries and coasts: spatial and temporal intercomparisons. Olsen & Olsen, Fredensborg, Denmark, pp. 375-384.
Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.
Smaal, A.C. and J. Widdows, 1994. The scope for growth of bivalves as an integrated response parameter in biological monitoring. In: K. Kramer (Ed.), Biomonitoring of coastal waters and estuaries. CRC Press, Boca Raton, pp. 247-268.
Van der Tol, M.W.M. and H. Scholten, 1992. Response of the Eastern Scheldt ecosystem to a changing environment: functional or adaptive? Neth. J. Sea Res. 30, 175-190.
Van der Tol, M.W.M. and H. Scholten, 1998. A model analysis on the effect of decreasing nutrient loads on the biomass of benthic suspension feeders in the Oosterschelde (SW Netherlands). Journal of Aquatic Ecology 31, 395-408.
Willows, R.I., 1992. Optimal digestive investment: a model for filter feeders experiencing variable diets. Limnol. Oceanogr. 37, 829-847.

9 Towards a model ontology instantiated for simulating bivalve ecophysiology

As far as the laws of mathematics refer to reality,
they are not certain,
and as far as they are certain,
they do not refer to reality.
(Albert Einstein, 1879-1955)

Content of Chapter 9

This chapter is based on parts of the following papers:

Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.

Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.

Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (*Cerastoderma edule* L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.

9.1 Introduction

The ontology presented in this chapter aims to describe (mathematical) models, with which the problems, discussed in Chapter 8, have to be solved. As part of the overall problem solving ontology (see Figure 9-1 and for more details Chapter 4, section 4.7 and figure 4.6) it uses the vocabulary of the meta-ontology (Chapter 5) and it should cooperate with the modelling ontology of Chapter 6 and the problem ontology of Chapter 8. Just as in Chapter 6 and Chapter 8, ontologies consist of an ontological structure and an ontological content. The latter is often called a knowledge base (KB).

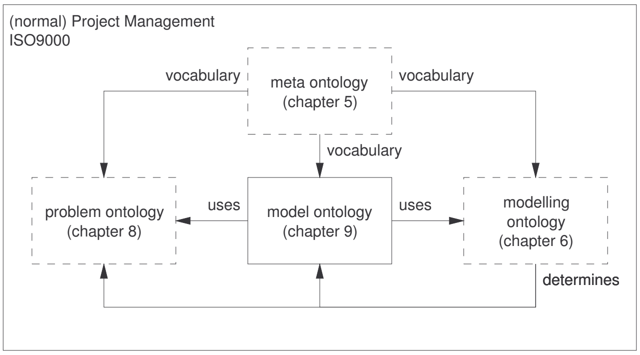


Figure 9-1. Linkage between the model ontology of this chapter and the various other ontologies.

This chapter aims at providing an ontology for *mathematical models*, belonging to different *model solving paradigms*, in order to prevent ambiguity or in other words to solve semantic problems about model concepts, which hinder cooperation between persons with different disciplinary backgrounds or with sometimes differing paradigmatic experiences. Some concepts are shared between models from different paradigms, while others are specific for a paradigm. The ontological structure proposed here should organize shared concepts together and concepts that are paradigm specific in a separate group. As 'proof of principle' the model ontology has been specified for a single model solving paradigm in detail, i.e. for continuous simulation models, and more specifically for bivalve ecophysiology.

Models were defined in the meta-ontology (Chapter 5) as:

> *To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.*

There are many types of models, but in this chapter we will only discuss *mathematical models*, using the description and definitions of Chapter 2.

This chapter will continue with a section on the design requirements for a model knowledge base (section 9.2) by proposing an ontological structure for models and indicate a classification of *concepts*. Section 9.3, the main part of this chapter, will attempt to present a structure for a (reusable) ontology for models. The first part after the introduction to the model ontology (9.3.1) describes *ontological layer 1* of the model ontology (section 9.3.2). The next section (9.3.3) provides concepts of the model ontology for *mathematical models*. In section 9.3.5 this will further be specialized for simulation models. Section 9.3.5 presents a model (simple) simulation model to exemplify how the concepts of the model ontology can be used to specify a (continuous simulation) model. In section 9.3.6 the ontological structure for bivalve ecophysiology models will be populated with instances. Section 9.3.7 describes some models for bivalve ecophysiology projects. Section 9.4 will evaluate the appropriateness of the proposed model ontology.

The model terminology and model content presented in this chapter are not new. They are based on commonly accepted scientific knowledge that was discussed in Chapter 8. The way of organizing this knowledge makes it new and facilitates its use and even enables reuse.

9.2 Design requirements for a model knowledge base

The design of the model knowledge base is a complex matter. It consists of a structure and the content of the KB. Like the process ontology (Chapter 6) and the problem ontology (Chapter 8), this structure should preferably be organized in *ontological layers*. Lower layers are more generic and easier to reuse. The more specific higher layers are more difficult to reuse, but contain the more practical knowledge. The content of the knowledge base is described in *instances*[1] of structural *concepts*[2]. The more structural formal parts of an ontology have to be – by definition[3] – manageable for *machine actor* and *human actors* and the instances (i.e. the content of the KB) should only understandable for *human actors*.

The major design requirement comprises what concepts and which relations between the concepts to include. Allocating the concepts in the right *ontological layers* is an extra requirement in our approach. In this way it becomes easier to reuse parts of the ontology. Here the following *ontological layers* will be used:

- Ontological layer 0: *meta-ontology* with basic terminology;
- Ontological layer 1: *generic model knowledge*;
- Ontological layer 2: *mathematical model knowledge*;
- Ontological layer 3: *simulation model knowledge*;
- Ontological layer 4: *(simulation) model knowledge for application domains*, here bivalve ecophysiological models[4];
- Ontological layer 5: *model knowledge for projects*, here for bivalve ecophysiological modelling projects.

Ontological layer 0 (meta-ontology) has been discussed in Chapter 5. *Ontological layer 1* (generic model knowledge), *ontological layer 2* (mathematical model knowledge) and *ontological layer 3* (simulation model knowledge) are the major goal of this chapter, as these are intended to be the easiest to reuse. *Ontological layer 4* (model knowledge for application domains, i.e. bivalve ecophysiological models) and *ontological layer 5* (model knowledge for projects, i.e. for bivalve ecophysiological modelling projects) are necessary to specify an example knowledge base to prove the usefulness of the whole ontological framework for models and especially for simulation models. Ontologies should be a body of knowledge, shared by a group and usable for a specific purpose. Only its use can prove its usability. Therefore, *ontological layer 2* (mathematical model knowledge) and *ontological layer 3* (simulation model knowledge) has to be tested by instantiating it for bivalve ecophysiological models (*ontological layer 4*) and associated projects (*ontological layer 5*). The classification in *ontological layers* for the model ontology is depicted in Figure 9-2.

Detailing *ontological layer 2* (mathematical model knowledge) and *ontological layer 3* (simulation model knowledge) in *ontological 4* (model knowledge for application domains, here bivalve ecophysiological models) will provide detailed simulation model content. The topic (bivalve ecophysiology) that has been chosen fits with the knowledge base topic of Chapter 8 (problem ontology) and its concreteness is reflected in the series of papers on bivalve ecophysiology models (Klepper and Scholten, 1988, Klepper, 1989, Klepper *et al*., 1994, Scholten *et al*., 1994, Smaal, 1997, Smaal and Scholten, 1997, Scholten and Smaal 1998, 1999, Rueda *et al*., 2005).

In contrast to the *project journals* for modelling projects (Chapter 6 and Chapter 7), no similar ontological pro forma[5] and associated tool for *ontological layer* 4 of the model ontology (*bivalve ecophysiological model used in projects*) have been developed.

[1] *Instance* is defined in the meta-ontology of chapter 5.
[2] *Concept* is defined in the meta-ontology of chapter 5.
[3] This definition is given in chapter 3: *an ontology is a concise and precise, formal specification, shared by a group of persons and providing sufficient vocabulary that a piece of knowledge can be formalized for its purpose, is understandable for its human users and manageable for its machine users (computers).*
[4] Meant are 'models' in all *model modes*, i.e. *conceptual model*, *mathematical model* and *computer model*, all defined in Appendix G, Table G-1.
[5] As stated in Chapter 8, a *model journal* or a *project journal* can be seen as a *pro forma* of what has been done in a modelling project or a process oriented project in general. A problem pro forma can best be integrated with in a *project journal* in order to keep all relevant pieces together and available and make projects more transparent.

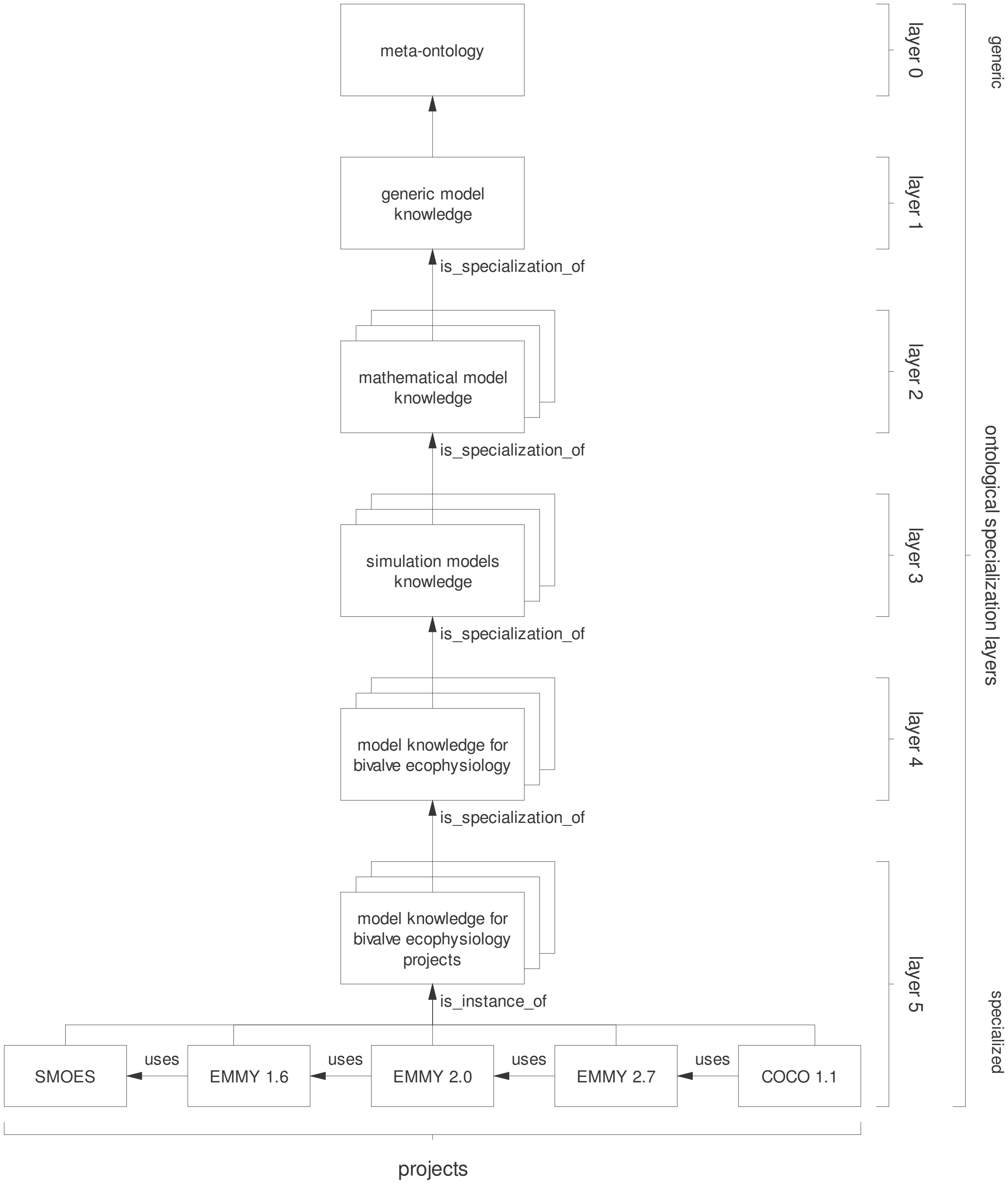


Figure 9-2. A stepwise ontology specialization of the model ontology. Solid rectangles are concepts that will be described in detail and dotted rectangles are not particularized.

9.3 A proposal for a model ontology

9.3.1 An appropriate ontological framework

This section outlines the ontological structure for models, and the associated knowledge about models. The *ontological layers 1* (section 9.3.2), *2* (section 9.3.3) and *3* (section 9.3.4) in this structure are the major parts and are meant to be reusable to a large extent. *Ontological layer 1* should be reusable for models in general and *ontological layer 2* should help to define *mathematical models*. *Ontological layer 3* is focused on simulation models. *Ontological layer 4* uses *ontological layers 2* and *3* and enables to define simulation models, here for bivalve ecophysiology by instantiating the concepts of *ontological layers 1, 2* and *3*. *Ontological layer 5* allows describing models used in projects (see Figure 9-2).

Just as was the case for the process ontology (Chapter 6) and the problem ontology (Chapter 8), the proposed framework with *ontological layer*s for the model ontology improves on the commonly adopted organization of ontologies. The framework with *concepts* and *instances* of *concepts* with the knowledge base content organized in *ontological layers* of increasing specialization facilitates extension of the KB or replacing parts that are reusable for other cases.

The framework with *ontological layer*s for the model ontology is very explicit and facilitates reuse, but allocating concepts to an *ontological layer* is again somehow arbitrary. Model concepts and their relations are allocated to *ontological layer*s according to Table 9-1. In *ontological layer*s *2*, *3* and *4* there are many alternatives, but in each layer only one alternative is instantiated. Using *ontological layer*s with increasing specialization to define model knowledge allows to be generic in *ontological layer*s *0* and *1* and to go into very specialized details in *ontological layer*s *4* and *5*. Therefore the content of *ontological layer 4* and *ontological layer 5* is only readable for a small group of persons that combine expertise in models and modelling with know-how in bivalve ecophysiology. The instances in *ontological layer 4* differ in the following way. In *ontological layer 4* modelling knowledge on continuous simulation models for bivalve ecophysiology is defined without connections to specific object system data. The actual data that changes a model definition in *ontological layer 4* into a concrete *site specific computer model* to simulate a specific *problem* in its *object system* belongs to *ontological layer 5*.

In the next sections (9.3.2, 9.3.3, 9.3.4, 9.3.6 and 9.3.7) the four *ontological layer*s of the model knowledge base will be presented. Each section consists of a *structure diagram* depicting the *concepts* and the *relations* between these *concepts* of that *ontological layer*. Definitions of all *concepts* are given in the tables of Appendix G.

Table 9-1. Allocation of concepts to the *ontological layers* of the model ontology.

Ontological layer	Summary
0. *Meta-ontology*	• Content: terminology required for other ontologies • Details: Chapter 5
1. *Generic model knowledge*	• Content: concepts, relevant for all models (OS), e.g. representation power (structural, behavioral), model objective, base model, model paradigm, etc. • Details: section 9.3.2
2. *Mathematical model knowledge*	• Content: concepts relevant for (quantitative) models, e.g. model mode (conceptual, mathematical, computer, site specific computer model), model scope, numerical model solver, objective function, model assumption, model component, model quantity (model variable type, model parameter type, model function type), model expression, model equations model input, model output, etc. • Details: section 9.3.3
3. *Simulation model knowledge*	• Content: concepts related to physical processes, e.g. observable variable, not-observable variable, differential equation, algebraic equation, model experiments, model scenario, experimental frame and model quantity related concepts (often specializations of concepts from layer 2) like: o *model variable type* (defined in *ontological layer 2*): e.g. *state variable, auxiliary variable*; o *model parameter type* (defined in *ontological layer 2*): e.g. *constant parameter*, *observed parameter*, *decision parameter*, *calculated parameter*, *free parameter*; o *model function type* (defined in *ontological layer 2*): e.g. *spatio-temporal series*, *spatial series*, *time series*, *tabular function*, *basic function*, *other function*; • Instantiated for: problems related to natural sciences, including

	environmental problems
	• Some alternative instantiations:
	  ○ Concepts for optimization models;
	  ○ Concepts for agent based models
	• Details: section 9.3.4

4. *Generic simulation models for application domains*	• Content: concepts specific for application domains, e.g. food related processes, respiration and excretion related processes, reproduction related processes, allocation and growth related processes.
	• Instantiated for: bivalve ecophysiology
	• Some alternative instantiations:
	  ○ A model to assess ecophysiology of other vertebrate or invertebrate organisms.;
	  ○ An ecosystem model.
	• Details: section 9.3.6

5. *Specific simulation models for projects*	• Content: Extension of the knowledge of ontological layer 4 for specific projects, i.e. a template for model characteristics.
	• Instantiated for: a series of models related to the ecophysiology of mussels and cockles.
	• Some alternative instantiations:
	  ○ A model project on the ecophysiology of oysters (Crassostrea gigas, Ostrea edulis);
	  ○ A model project on the ecophysiology of cut trough shell (Spisula subtruncata).
	• Details: section 9.3.7

9.3.2 Ontological layer 1: generic model knowledge

Ontological layer 1 will be filled with *concepts* and the *relations* connecting them, which are common for all kind of *models*. The *concepts* and *relations* in this *ontological layer* should be instrumental for discussing the various concepts in *ontological layers 2*.

The concepts of *ontological layer 1* (generic model knowledge) are defined in Appendix G (Table 9-1). All concepts and relations of this *ontological* layer are depicted in the *structure diagram* of Figure 9-3 . The concepts in *ontological layer 1* include generic model terminology, as is summarized in Table 9-1.

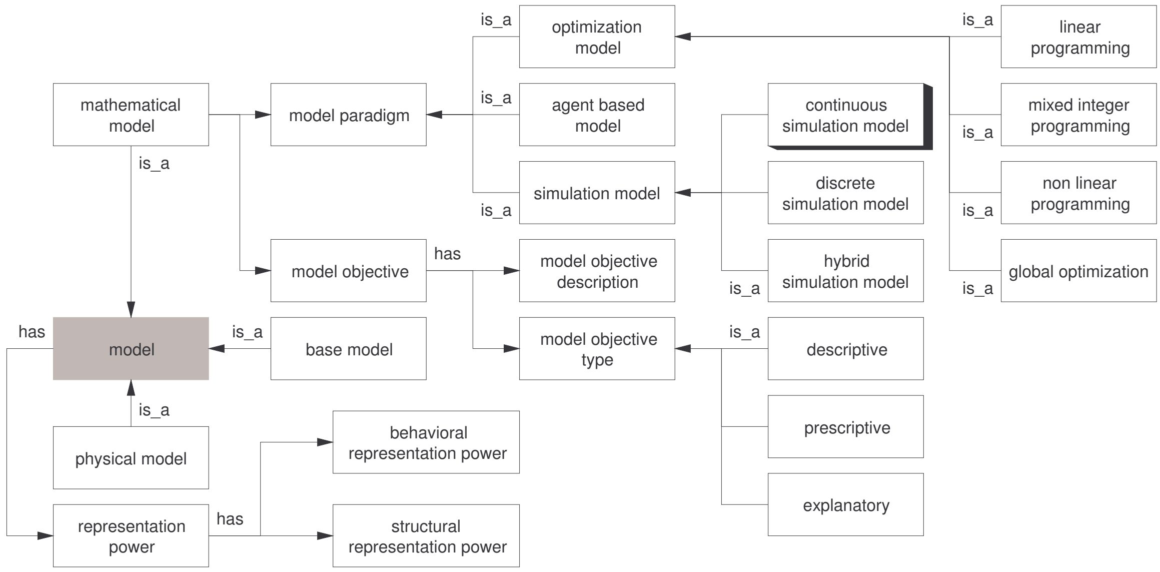


Figure 9-3. *Detailed structure diagram of ontological layer 1 (*generic model knowledge). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0* (meta-ontology of Chapter 5) and shadowed *concepts* are detailed in *ontological layer 2* (section 9.3.3). Some *concepts* and *properties*, defined in Appendix G, Table 9-1 are left out.

9.3.3 Ontological layer 2: mathematical model knowledge

Ontological layer 2 is filled with *concepts* and the *relations* connecting them, which are specific for *mathematical models*. The terminology, i.e. *concepts* and *relations* in this layer should be instrumental to define and discuss simulation models for a class of *mathematical models*. *Ontological layer 3* specializes further to (continuous) *simulation models*. In *ontological layer 4* the concepts of *ontological layer 3* will be instantiated for a generic bivalve ecophysiology model. Some concepts originate from the development of Smart, Simulation and Modelling Assistant for Research and Training (Kramer and Scholten, 2001).

The definitions of the *concepts* of *ontological layer 2* are given in Appendix G (Table G-2) and Figure 9-4 depicts the associated *structure diagram* of *ontological layer 2*. Together they represent *ontological layer 2* (mathematical model knowledge).

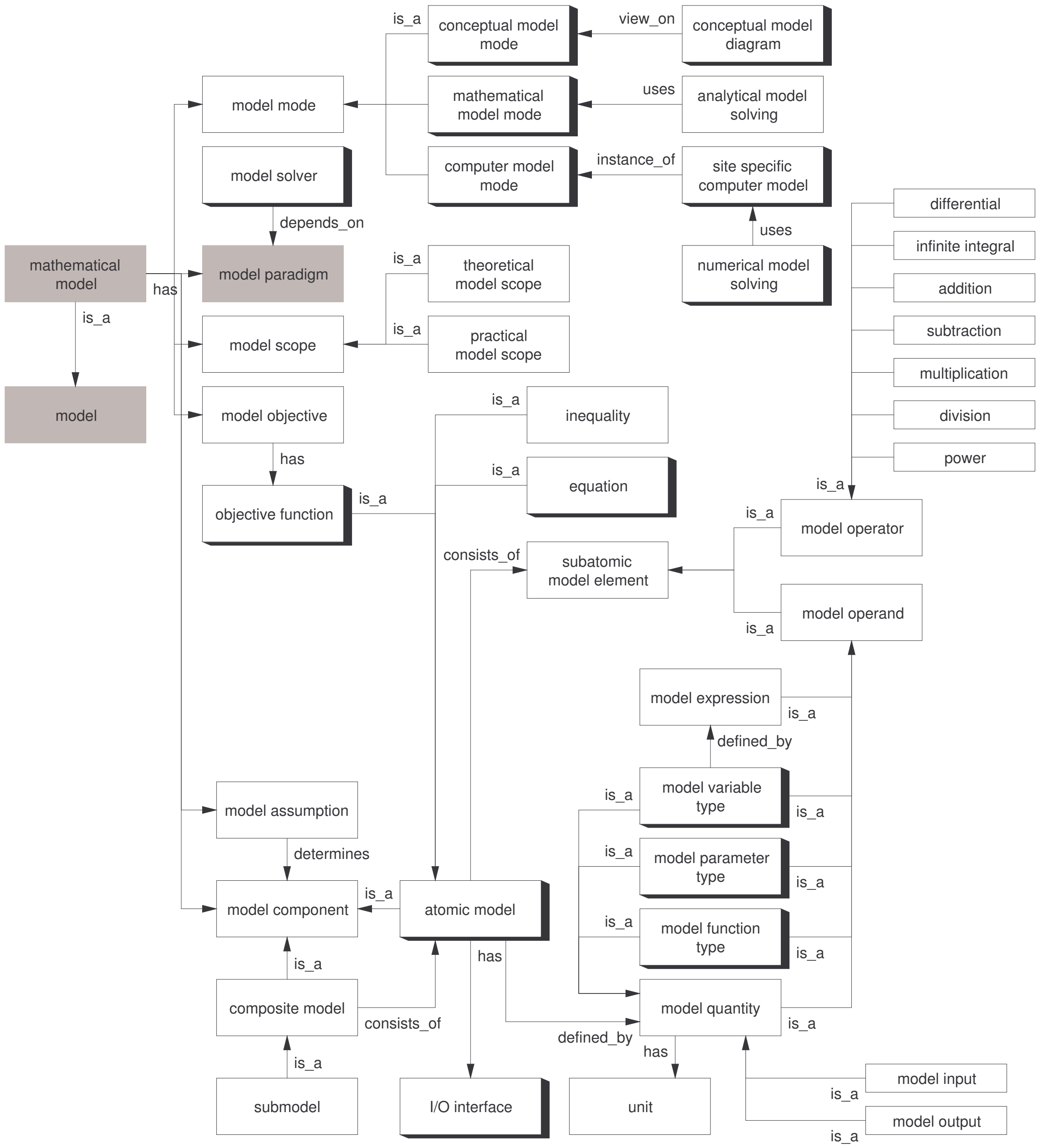


Figure 9-4. Detailed *structure diagram* of *ontological layer 2* (mathematical model knowledge). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0* (*meta-ontology* of Chapter 5) or *ontological layer 1*. Shadowed *concepts* are detailed in *ontological layer 3* (section 9.3.3). Some *concepts* and *properties*, defined in Appendix G, Table 9-2 are left out.

9.3.4 Ontological layer 3: simulation model knowledge

Ontological layer 3 is filled with *concepts* and the *relations* connecting them, which are specific for *(continuous) simulation models*. The *concepts* and *relations* in this layer should be instrumental to define *simulation models* in *ontological layer 4* . This will be done for a generic bivalve ecophysiology model. Some concepts of *ontological layer 4* originate from the development of Smart, Simulation and Modelling Assistant for Research and Training (Kramer and Scholten, 2001).

The definitions of the *concepts* of *ontological layer 3* are given in Appendix G (Table G-3) and Figure 9-5 depicts the associated *structure diagram* of *ontological layer 3*. Together they represent *ontological layer 3* (simulation model knowledge).

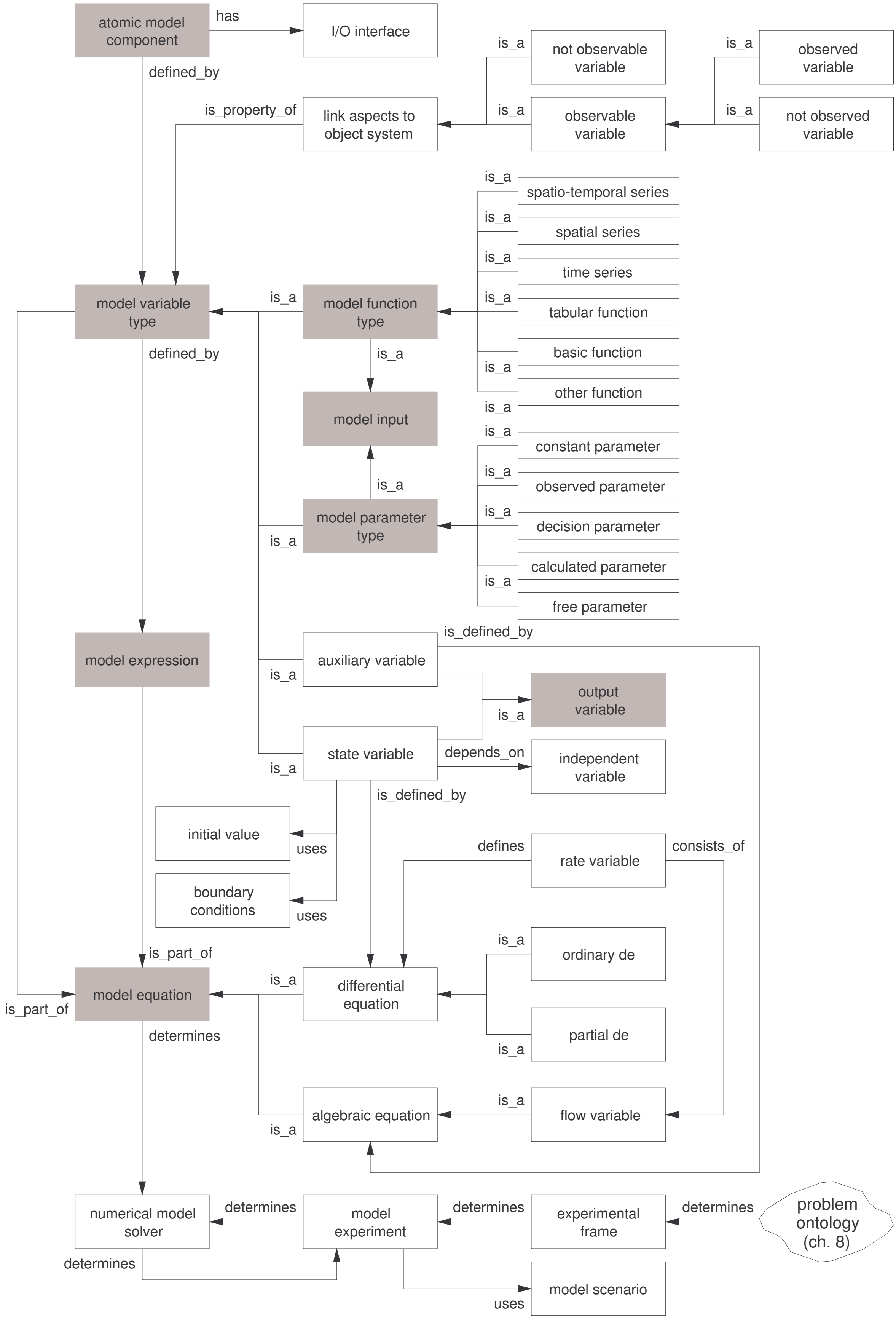


Figure 9-5. Detailed *structure diagram* of *ontological layer 3* (simulation model knowledge). Rectangles are *concepts* and arrows *relations*. Grey *concepts* are defined in *ontological layer 0* (meta-ontology of Chapter 5) and *ontological layer 1* and *2*. Some *concepts* and *properties*, defined in Appendix9, Table 9-3 are left out.

9.3.5 Example model: Lynxes and Snowshoe Hares

To explain the use of some of the terms defined in Appendix G (Table G-2) an example of a conceptual simulation model and the corresponding mathematical simulation model are shown. This model is based on a classical set of data on a pair of interacting populations that come close: the Canadian lynx and snowshoe hare pelt-trading records of the Hudson Bay Company over almost a century.

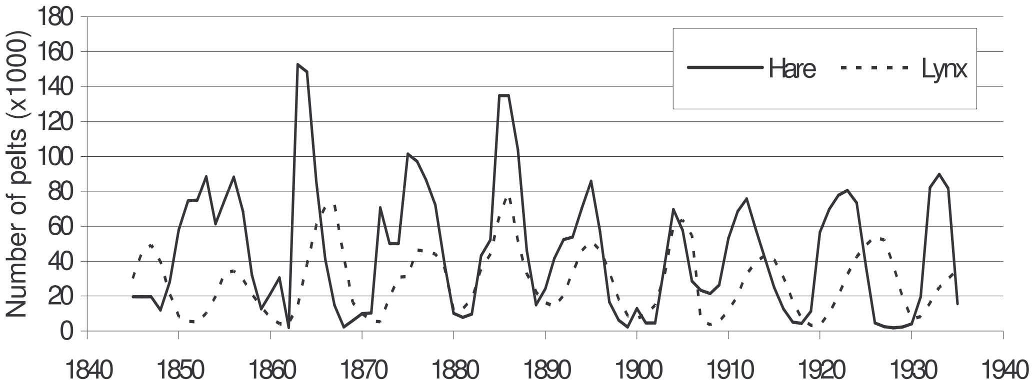


Figure 9-6. Plot of the data of interacting populations of Canadian lynx (*Lynx canadensis*) and snowshoe hare (*Lepus americanus*), based on pelt-trading records of the Hudson Bay Company over almost a century (adapted from Odum, 1953)

A model that aims to describe the interaction between a predator species (*Lynx Canadensis*, Canadian lynx) and a prey population (*Lepus americanus*, snowshoe hares) is the classic Lotka-Volterra model (Lotka, 1925, Volterra, 1926). The model version, used here as an example, is called Lynxes and Snowshoe Hares (LandSH).

The following *model assumptions* have to be made:

- The prey has unlimited resources.
- The prey's only threat is the predator.
- The predator is a specialist; i.e., the predator's only food supply is the prey.
- The predator's growth depends on the prey it catches.
- The prey's mortality is solely caused by the predator.

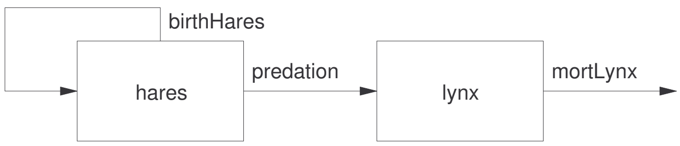


Figure 9-7. A *conceptual model diagram* of LandSH. Rectangles are *state variables* an arrows are *flow variables*.

Table 9-2. Mathematical model LandSH, based on the conceptual model depicted in Figure 9-7.

$$\frac{dx}{dt} = ax - bxy$$

$$\frac{dy}{dt} = pxy - cy$$

with

x – state variable representing the number of snowshoe hare
y – state variable lynx representing the number of lynx
a – specific birth rate snowshoe hare
b – efficiency of the lynx' ability to capture hares
c – specific mortality rate of lynx
p – efficiency of lynx to increase in numbers by eating snowshoe hares

Table 9-3. The *mathematical simulation model* of LandSH as is implemented as in Smart[6], based on the conceptual model depicted in Figure 9-7 and the mathematical model of Table 9-2. 'Initial values' is not a concept of *ontological layer* 3, but to *ontological layer 5*. '#' means 'number of'.

Derivative of state variable		Rate variable	Unit	Meaning
d(Hare)/dt	=	birthHare - predationHare	[#hares]	Number of hares
d(Lynx)/dt	=	predationLynx - mortalityLynx	[#lynx]	Number of lynx
Initial value		**Value**	**Unit**	
$Hare_{t=0}$		1000	[#hares]	Initial values hares
$Lynx_{t=0}$		20	[#lynx]	Initial value lynx
Flow variable		**Definition of flow variable**	**Unit**	**Meaning**
birthHare	=	hare*SpBirthRateHare	[#hares.y^{-1}]	Birth of hares
predationHare	=	AttackRate*hare*lynx	[#hares. y-1]	Predation of hares
predationLynx	=	Attack_Rate*ConversionHareToLynx *hare*lynx	[#lynx.y-1]	Predation of hares conversed to birth of lynx
mortalityLynx	=	lynx*SpMortRateLynx	[#lynx.y-1]	Mortality of lynx
Parameter		**Value**	**Unit**	**Meaning**
SpBirthRateHare	=	0.25	[y^{-1}]	Specific growth rate of hares
AttackRate	=	0.01	[$lynx^{-1}.y^{-1}$]	Efficiency of the lynx' ability to capture hares
ConversionHareToLynx	=	0.008	[$lynx.hares^{-1}.y^{-1}$]	Conversion efficiency hare → lynx
SpMortRateLynx	=	0.1	[y^{-1}]	Specific mortality rate Lynx

9.3.6 Ontological layer 4: (simulation) model knowledge for application domains, instantiated for bivalve ecophysiology models

9.3.6.1 Introduction

In this specialized *ontological layer 4,* knowledge on continuous simulation models for bivalve ecophysiology will be organized and instantiated in the concepts of *ontological layer 3*. Most concepts and other terminology related to the problem and object system, i.e. the content of the model, are explained in Chapter 8, especially section 8.3.4). The instances of the concepts of *ontological layer 3* in *ontological layer 4* should facilitate the discussion of this type of models and enabling to generate new ones. Other possible instances – not specified here – can contain knowledge for other (generic) continuous simulation models, e.g. ecosystem models, air pollution models, crop growth models, etc.

The knowledge for this model is derived from the mussel model EMMY (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999) and the cockle model COCO (Rueda *et al.*, 2005). First the conceptual model will be presented, which is reconstructed based on the three applications of the EMMY model[7] and on the COCO model. Subsequently a full description will be given of the mathematical model. This is mainly based on the formulations of Scholten and Smaal (1999) and differences with Rueda *et al*. (2005) will be indicated. Beadman *et al*. (2002) give a review of ecophysiological models for mussels.

9.3.6.2 Conceptual model of bivalve ecophysiology

A more complete description of the scientific knowledge related to the ecophysiology of bivalves is discussed in Chapter 8. Here a summary should help to understand translation of this knowledge into a conceptual model (see Figure 9-8).

If *bivalves* are submerged, water with *seston* surrounds them. Seston consists mainly of (inorganic) silt with no nutritional value at all in many systems and of living *phytoplankton* with high nutritional value (and high protein content, measured as nitrogen), fresh *detritus* (recently deceased *phytoplankton*, also known as *labile detritus*) and old *detritus* with low protein content and almost no nutritional value, also known as *refractory detritus*. *Phytoplankton* and both forms of *detritus* are *food* for *bivalves* (and many other organisms). Food

[6] See Kramer and Scholten, 2001.
[7] Each subsequent application has been based on improved versions of EMMY. COCO is a version of EMMY, adapted for cockles (instead of mussels) by leaving out all nitrogen state variables and processes. Furthermore, food intake and activity related respiration (*ResRout*) occurs only during submersion.

concentrations vary in the water column from high concentrations near the bottom (where bivalves live) to low concentrations near the water surface.

Bivalves pump water (*clearance*) along their *gills*. Clearance can be depressed in case of (1) high silt content of the seston, (2) high concentrations of Phaeocystis-colonies (these colonies clog the *gills*) or (3) a pure *phytoplankton* diet. The *gills* remove a part of the *seston* by *filtration*. Here bivalves can select better parts of the seston by *pre-ingestive selection* in order to enrich the quality of their food. What they select is *ingested* and enters the gut, consisting of *stomach* and *intestines*. What is not ingested is packed in *mucus* and ejected as *pseudofaeces*. The ingested material, a mix of food and silt stays in the gut for a time that depends on the food availability and quality. During that digestion time a (larger) part of the organic material is *absorbed* through the wall of the *gut* and can be used by the *organism* for all kind of purposes.

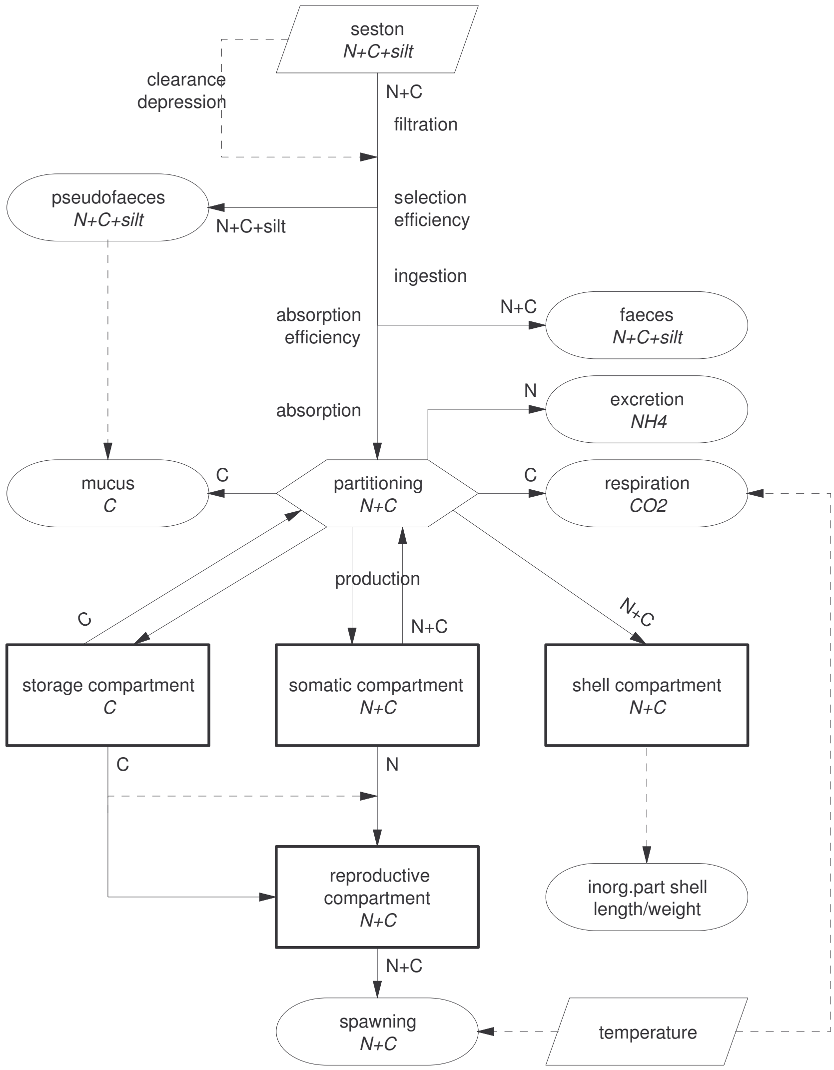


Figure 9-8. Conceptual model diagram of a *continuous simulation model* for *bivalve ecophysiology*, including the EMMY model for *mussels* and the COCO model for *cockles*. Rectangles are *state variables*, parallelograms are *input time series*, ovals are *flow variables* leaving the system and arrows are intermediate *flow variables*, which have a meaning within the system. 'C' refers to carbon and 'N' to nitrogen. *Processes* (*flow variables*) related to nitrogen are assumed to be relevant for mussels and irrelevant for cockles. For the latter species *excretion* is not included in the model. The figure is a reconstruction based on Smaal and Scholten (1997), Scholten and Smaal (1998, 1999) and Rueda et al. (2005).

Maintenance is the main purpose of the absorbed material. Maintenance consists of *respiration* (producing energy by assimilating organic material) and *excretion* of a surplus of nitrogen (in the form NH_3) to keep a relatively constant C/N-ratio. If there is more *absorbed* material than is needed for maintenance the rest will be allocated to the *somatic compartment*, the *storage compartment*, the *reproductive compartment* and the *organic shell compartment*. If *absorbed material* cannot accommodate *maintenance* requirements, i.e. (organic) carbon needed as food source for *respiration*, the deficit will first be compensated from the *storage compartment*, which consists of the carbohydrate *glycogen*. If the *storage compartment* is not sufficient, the rest will be balanced by resorption of structural body tissues, i.e. the *somatic compartment*. The latter indicates starvation and can lead to the death of the organism, if continued too long. Nitrogen shortage will not occur easily, only if the energy demand (respiration) is provided from the storage compartment for a substantial period.

In the Netherlands *reproduction* occurs mainly once a year, if several conditions are met, i.e. sufficient relative size of *gonads* (the reproductive organs with reproductive cell, the *gametes*) and a minimum water temperature. The release of the *gametes* is called *spawning* and *bivalves* can spend 10-30% of their total *bodyweight* in this reproductive process.

9.3.6.3 Mathematical model of bivalve ecophysiology

The *conceptual model* of section 9.3.6.2 has been translated into a *mathematical model* (see Appendix G, Table G-4, Table G-5, Table G-6, Table G-7, and Table G-8). This *mathematical model* has subsequently been implemented in a *computer model* that can be *analyzed numerically* with a proper *numerical model solver*. A better understanding of the mathematical model can be obtained by comparing the mathematical formulations with the knowledge of Chapter 8 (section 8.3.5) or in summary in section 9.3.6.2.

9.3.7 Ontological layer 5: model knowledge for projects

9.3.7.1 Template

The knowledge of *ontological layer 4* has been used in several model applications, all aiming at simulating bivalve ecophysiology. Ontological layer 5 consists of a structural part (this section) and instantiations of this structural part (section 9.3.7.2). The concepts presented in this structural part are rather generic, as they act as a template to describe models used in projects on bivalve ecophysiology and ecological processes. Appendix G (Table G-9) gives an overview of the concepts in *ontological layer 5*. All concepts are self-explaining characteristics of (scientific) projects and will therefore not be discussed here.

9.3.7.2 Instances of bivalve ecophysiological model applications

9.3.7.2.1 Simplified ecophysiology in SMOES

The ecosystem model SMOES has been developed and used in the context of two (large) model application projects. In this *ontological layer 5*, i.e. instances of the model project template of Appendix G, Table G-9, characteristics of the model application of SMOES are summarized. SMOES has been used in two larger ecosystem studies in the eighties and nineties of last century (Klepper and Scholten, 1988, Klepper, 1989, Klepper et al., 1994, Scholten et al., 1994). The bivalve submodel of SMOES is used as starting point to develop the mussel model EMMY (section 9.3.7.2.2) and the cockle model COCO (section 9.3.7.2.3). The model characteristics of SMOES are described in Appendix G, Table G-10.

9.3.7.2.2 EMMY: an ecophysiological model of *Mytilus edulis* L.

The mathematical model of bivalve ecophysiology of Appendix G (Table G-4, Table G-5, Table G-6, Table G-7 and Table G-8) has been used in three versions for mussels and in another (fourth) version for cockles. All mussel versions are called EMMY.

The first published model version (version 1.6, Smaal and Scholten, 1997) has been developed to simulate the growth of a single mussel under various ecological conditions and was tested with the Oosterschelde data for the period 1982-1987 (Smaal, 1997, Smaal and Scholten, 1997). This model version described rather well the mussel growth data that have been collected in the BALANS and EOS project (Nienhuis and Smaal, 1994). A major difference with the mathematical model of bivalve ecophysiology, described in Appendix G (Table G-4, Table G-5, Table G-6, Table G-7, and Table G-8) is an extra *state variable* 'blood' that played a major role in the allocation of food not needed for *maintenance* to the other *state variable* representing the functional compartments (*somatic compartment*, *storage compartment*, *reproductive compartment* and *organic shell compartment*). The model characteristics of EMMY, version 1.6 are described in Appendix G, Table G-11.

The second published version of EMMY (version 2.0, Scholten and Smaal, 1998) had to cope with a wider range of food concentrations and was tested for natural conditions in three ecosystems: the Oosterschelde estuary (SW Netherlands), the bay of Marennes-Oléron (France) and Upper South Cove (Canada). This version was more or less equal to the first published version of Smaal and Scholten (1997). The model characteristics of EMMY, version 2.0 are described in Appendix G, Table G-12.

The third published version of EMMY (version 2.7, Scholten and Smaal, 1999) has been simplified compared to earlier versions, leading to the generic mathematical model of bivalve ecophysiology, described in section 9.3.6. The simplifications consisted of leaving out the state variables ‘bloodC’, ‘bloodN’, ‘spawnC’, ‘spawnN’, which were not necessary to obtain a model behavior similar to the knowledge as described in Chapter 8. The model characteristics of EMMY, version 2.7 are described in Appendix G, Table G-13.

The three versions of EMMY that were used for the publications, i.e. Smaal and Scholten (1997), Scholten and Smaal (1998) and Scholten and Smaal (1999), were step-wise improvements. The last, most simplified version 2.7, is still complex compared to other models (Beadman *et al*., 2002), but this version was able to show mussel growth comparable to what has been observed and measured in similar conditions. It can deal with a wide range of ecological conditions and is therefore an adequate model for mussel growth, as mussels are also able to handle extreme conditions.

9.3.7.2.3 COCO: an ecophysiological model of *Cerastoderma edule* (L.)

COCO, an ecophysiological model of *Cerastoderma edule* (L.) has been based on EMMY 2.0, which is a version without state variable ‘blood’, but with state variable SpawnC. All *nitrogen* related *state variables*, *processes* and other *variables* were left out, as being irrelevant. This can be explained as follows. The cockles described by the model live on intertidal flats and can easily remove a surplus of nitrogen during low tides. An extra parameter has been added to control the effect of high and low tides by determining submersion as fraction of the day. The instance of COCO that has been used by Rueda et al. (2005) is summarized in Appendix G, Table 9-14.

9.4 Scope and appropriateness of the proposed model ontology

Knowledge bases are hard to evaluate and also determining the scope and the appropriateness of the proposed model ontology is not easy. The setup in *ontological layer*s (section 9.2 and Figure 9-2) allows discussing the scope of each ontological separately.

Ontological layer4 and *ontological layer 5* structure and instantiate real projects, which shows their appropriateness. That leaves *ontological layer*s *1*, *2* and *3* to be assessed.

Distributing concepts over these three *ontological layer*s seems straightforward, but distinguishing generic model knowledge (*ontological layer* 1) from mathematical model knowledge (*ontological layer 2*) and (continuous) simulation model knowledge (*ontological layer 3*) is not easy at all. *Ontological layer 1* should be rather generic and of use for many types of models and other decision supporting instruments, including - but not restricted to - optimization models. Reuse of *ontological layers 1*, *2* and *3* should be easy and constructive for practical purposes, but this has not been tested as yet.

The content of *ontological layer 4* has been tested in several practical applications, i.e. for mussel ecophysiological models (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999) and for a cockle ecophysiological model (Rueda *et al.*, 2005). A further test can be reusing *ontological layer 4* in the development of other bivalve ecophysiological models, e.g. oysters (*Crassostrea gigas*, *Ostrea edulis)* and the ‘cut trough shell’, *Spisula subtruncata*. Both candidates have commercial interest and the relevance of such models is emphasized by the interest of designing a culture, which is friendlier for nature and environment and more sustainable.

Testing the usability of *ontological layer*s *1*, *2* and *3*, of the model ontology can best be done by using these to build a knowledge base on completely other models and other decision supporting instruments, but there are no plans to do so in the near future.

9.5 References

Beadman, H.A., R.I. Willows and M.J. Kaiser, 2002. Potential applications of mussel modelling. Helgoland Marine Research 56, 76-85.

Klepper, O. and H. Scholten, 1988. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (SW Netherlands), DGW/DIHO, Middelburg, 278 pp.

Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary, DGW-Ministery of Transport and Public Works. Wageningen University, Wageningen, 270 pp.

Klepper, O., Van der Tol, M.W.M., Scholten, H., Herman, P.M.J., 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.

Kramer, M.R. and H. Scholten, 2001, The Smart approach to modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering., Delft, The Netherlands, June 26-29, 2001, TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0.

Lotka, A.J., 1925. Elements of physical biology. Williams and Wilkins, Baltimore.

Nienhuis, P.H. and A.C. Smaal (Eds.), 1994. The Oosterschelde estuary. Developments in Hydrobiology 97, H.J. Dumont, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-2817-5, Reprinted from Hydrobiologia, vols. 282/283, 597 pp.

Odum, E.P., 1953. Fundamentals of Ecology, W. B. Saunders, Philadelphia.

Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (Cerastoderma edule L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.

Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.

Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.

Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.

Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.

Smaal, A.C., 1997. Food supply and demand of bivalve suspension feeders in a tidal system, Wiskunde en Natuurwetenschappen. Rijksuniversiteit Groningen, Groningen, 237 pp.

Volterra, V., 1926. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. Mem. R. Acad. Naz. Lincei 2, 31-113.

10 On testing the proposed ontological framework for multidisciplinary model-based problem solving

Debugging is twice as hard as writing code in the first place.
Therefore, if you write the code as cleverly as possible, you are,
by definition, not smart enough to debug it.
Brian W. Kernighan (1942)

Content of Chapter 10

10.1 Introduction to the evaluation of the proposed ontological framework

According to many authors – an overview is given by Dadkhah and Abdollahzadeh Barfouroush (2004) – testing can be divided in two facets: verification and validation. *Verification* of a knowledge-based system (KBS) is the task of determining that the system is built according to its specifications. *Validation* is the process of determining that the system actually fulfils the purpose for which it was intended. Do the intended end-users adopt it and evaluate its performance positively. Verification is showing the system is built right and validation is showing the right system was built. However, these terms are rather ambiguous, as they are used in several disciplines in a different way with different definitions.

Testing is one of the most difficult tasks to do. What a test includes depends strongly of what you are testing. The main categories of things to test are processes and products. Here even the process ontology (instantiated for modelling) is a product and not a process. Therefore testing the ontological framework focuses on checking a product. Such a test has to be designed depending on the type of product. In science testing often consists of experimenting in order to falsify a theory or hypothesis; as soon as an experiments proves that a theory or hypothesis is wrong, the theory or hypothesis has to be replaced by a new one that can stand all the tests the previous theory or hypothesis could endure plus the (last) test it could not handle properly (Popper, 1959). Verification and validation of simulation models of natural systems is impossible (Oreskes *et al*., 1994). If such models pass a 'validation test', this should – at best - be seen as a part of a confirmation process of the model at hand. Refsgaard and Henriksen (2004) review this terminology in order to provide a concise and practical approach in model analysis[1]. But even with (some) negative results of a validation test a model can still be useful (stated by many authors, including Scholten and Van der Tol, 1994a, 1994b, 1998, Scholten *et al*., 1998). The evaluation of ontologies, and knowledge bases built on these ontologies, is to some extent similar to methods and techniques used to test the design and implementation of software (Preece, 2001). Before discussing how the ontological framework presented in this book and its associated tools are tested, testing of this kind of artefacts will be defined.

In Chapter 6 verification and validation in simulation modelling are discussed, but this approach cannot be copied to evaluating knowledge bases. Verification and validation are also a hot topic in software engineering, where they accompany testing of software. Next to the definitions for verification and validation in the ISO 9000 quality standards, the Institute of Electrical and Electronics Engineers gives definitions for these terms (IEEE, 1998)[2]. Although these definitions are very useful and intensively used, they can only be instrumental for the software part of the proposed framework (MoST/ProST and the KB-editor). In the communities of artificial intelligence, expert systems and knowledge based systems there are – in general – two approaches, the first more theoretical and formal and the second more practical and application oriented (Bench-Capon *et al*., 1999). The formal approach is not possible, because the proposed framework is a mixture of structural knowledge elements, knowledge content and software tools. In the knowledge engineering community working with ontologies, sometimes the approach proposed by Gómez-Pérez (2004) is followed, who uses the following definitions of verification, validation and assessment:

- *Ontology verification* refers to building the ontology correctly, that is, ensuring that its definitions implement correctly the ontology requirements and competency questions, or functions correctly in the real world.
- *Ontology validation* refers to whether the ontology definitions really model the real world for which the ontology was created. The goal is prove that the world model (if it exists and is known) is compliant with the world modeled formally.

---

[1] I use the term *model analysis* as a container term for *model confirmation*, *model code verification*, *model calibration* and *model validation*.

[2] IEEE Std 1012-1998 defines *verification* as *Confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled* and *validation* as *confirmation by examination and provisions of objective evidence that the particular requirements for a specific intended use are fulfilled*. IEEE Std 1012-1998 continues with *Software verification and validation (V&V) processes, which determine whether development products of a given activity conform to the requirements of that activity, and whether the software satisfies its intended use and user needs, are described. This determination may include analysis, evaluation, review, inspection, assessment, and testing of software products and processes. V&V processes assess the software in the context of the system, including the operational environment, hardware, interfacing software, operators, and users*. This standard is replaced by IEEE 1012-2004.

- *Ontology assessment* is focused on judging the ontology content from the user's point of view. Different types of users and applications require different means of assessing an ontology.

Furthermore, Gómez-Pérez (2004) gives criteria for evaluation of ontologies. In this Chapter ontology verification, ontology validation and ontology assessment will not be used. The activities evaluate the proposed framework will use a series of criteria (partly overlapping with those of Gómez-Pérez (2004) and tests for those aspects of the proposed framework that are ontological, but use also software engineering criteria and tests for the software components of the proposed framework. These tests and criteria can be classified in terms of verification (focused on following requirements and internal completeness and consistency) and validation (focused on usefulness and appreciation of users). This classification will not be used here. The proposed framework will be evaluated by *testing* what has been realized compared with predefined *criteria*, using a series of *tests*. Testing is a continuous process: each test aims at proving something is not 'good'; the more tests passed successfully, the more confidence one can have.

The next section of this chapter discusses the intensity with which each component of the proposed ontological framework has been tested. Subsequently, testing of each leaf on the ontological tree (meta-ontology, processes, problems, models) will be discussed. This chapter will end with conclusions based on testing the ontological framework. As the ontology for processes (including modelling) is further developed and used, it will automatically also be more intensely tested. For each ontological leaf the following aspects will be presented: which tests are available, which criteria should be checked, which tests are used for each criterion and what the results are of these tests.

10.2 Testing the proposed ontological framework

The proposed ontological framework consists of a meta-ontology, providing basic terminology for the rest of the framework, a process ontology, a problem ontology and a model ontology. The process ontology was developed within the HarmoniQuA project[3] and instantiated for the modelling process with emphasis on water management (Chapter 6). A Modelling Support Tool (MoST) accompanies the process ontology and this tool is also used for other processes, i.e. the water stress mitigation process, in the AquaStress project. Many of the side branches at the bifurcations (see Chapter 4, Figure 4-6 and Figure 10-1) are not instantiated (yet) and can therefore not be tested. The overall test efforts and their results are therefore limited and by no way a complete 'proof' of the proposed ontological framework, i.e. they do not provide evidence that the proposed framework is correct (without errors) nor that it provides the best functionality for the purpose it is designed for. If the tests have a positive result they just 'confirm' that the framework matches the criteria tested. In this way the testing of the framework has to be seen as a step in the confirmation process, as proposed for model analysis by Oreskes *et al*. (1994).

10.3 Testing the meta-ontology

The meta-ontology proposed in Chapter 4 does not aim at being of use for other purposes or reuse (see also Figure 10-1). It has been designed to act as a container for basic terminology to discuss and define the other ontologies of the ontological framework (on *modelling*, on *problem/object system* and on *models*).

Testing the meta-ontology has been approached in the simplest way. As this top leaf of the proposed ontological framework aims at providing basic terminology, the only requirements for this ontology are:

1. It should include all terms that need explanation or that are used in a special (non-trivial) way;
2. It should contain only those terms that do not belong to one of the other tree leafs of the ontological framework

This requirement can also serve as criterion in testing. It is assured that the meta-ontology meets this criterion along two lines. Firstly, terms have been added to the meta-ontology only if needed, i.e. when already used in Chapter 4 and when not belonging to another ontology. Subsequently, the meta-ontology is scrutinized in a face evaluation[4]. No essential problems have been observed over a period of time.

---

[3] HarmoniQuA contract EVK1-CT2001-00097, see www.HarmoniQuA.org/.

[4] Here 'face evaluation' is used instead of 'face validation'. Sargent (1984) reviews validation techniques in modelling and defines *Face evaluation* as 'asking people knowledgeable about the system whether the model and/or its behaviour are reasonable. This technique can be used in determining if the logic in the conceptual model is correct and if a

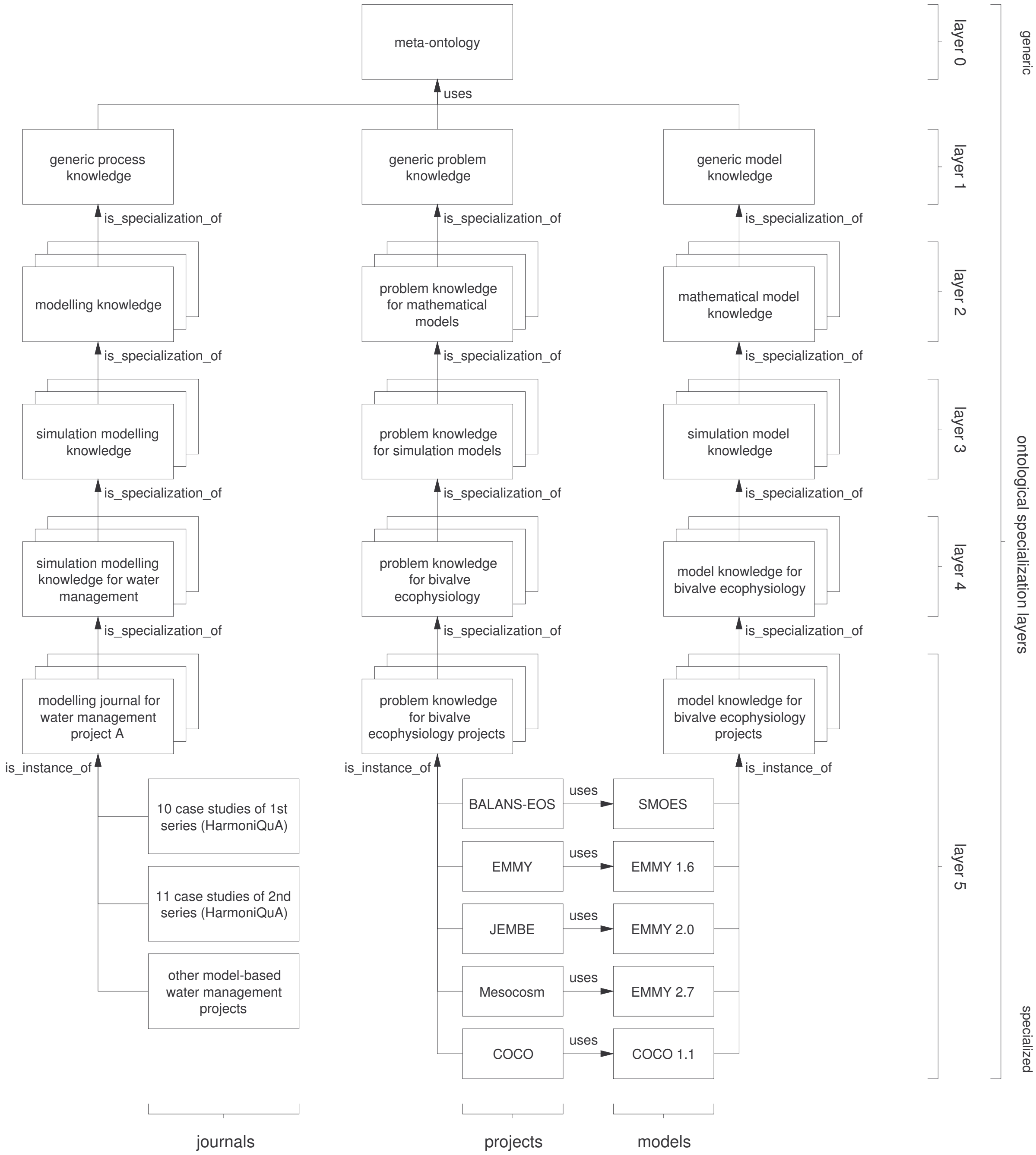


Figure 10-1. Simplified view of the knowledge part of the proposed ontological framework with stepwise *specialization* from the top (meta-ontology) to the *concepts* at the bottom. The three main branches are the *process ontology* (left), the *problem and object ontology* (central) and the *model ontology* (right). A set of three *concepts* behind each other indicates a bifurcation in the tree. The side branches at the bifurcations are left out in the next (more specialized) *ontological layer*.

10.4 Testing the modelling ontology and Modelling Support Tool

10.4.1 Test approach

Knowledge based systems are hard to test, as several aspects of these systems have to be evaluated. For the scientific cooperation process ontology and its instance for modelling different aspects are tested against various criteria and using a range of methods. These aspects include: the *KB structure*, the *KB process decomposition*, the *KB content*, the tool *MoST*, the *Training material + Help* and the *KB technology*.

model's input-output relationships are reasonable.' However, this technique has hardly anything to do with 'validation' and therefore the term 'face evaluation' is used here to test an ontology.

Next to the test cases many other tests have been applied. Table 10-1 gives an overview of these tests. A major test method used to test the modelling instance of the ontology and the modelling support tool MoST consists of case studies. MoST and its modelling knowledge base for water management have been developed in three phases, resulting in (1) an incomplete prototype providing basic guidance and monitoring, (2) a full version with almost all intended functionality and (3) a final version. The incomplete prototype has been tested in the first series of (10) case studies, which were real, commercial projects. The full version, developed in the second phase, has been tested in the second series of (11) test cases. The results of each test case series have been used to improve the next version of MoST resulting in the final version that has to be seen as a proof of principle. The full lists of case studies, the responsible HarmoniQuA partner and the water management domains included are listed in Table 10-2.

The test methods used (see section 10.4.2, Table 10-1) will be combined with the list of test criteria (see section 10.4.3, Table 10-6) to generate a list of how each test criterion has been tested and with which results (see section 10.4.4 and Appendix H, Table H-1).

10.4.2 Types of tests

10.4.2.1 Overview of tests used

Table 10-1 gives an overview of the tests used for the process ontology and explains each test. All tests have been planned in the HarmoniQuA research proposal except for 'reuse'.

Table 10-1. Test types used to test the *modelling ontology* and *MoST*. The test method names will be used in other tables.

Name	Explanation
Project discussion	*Discussed by all project partners together in the HarmoniQuA project.*
Internal KB	*Test by three project partners, not involved in developing the KB by reading and using.*
Internal MoST[5]	*Test by a single project partner, not involved developing MoST. This testing has been executed according to a test plan (Rocha, 2002).*
Changing ontology	*Changing the ontological structure tests how flexible the knowledge based system (MoST and its KB) is, i.e. how much effort is needed to adapt the KBS to changes in the ontology?*
Case studies	*Two series of case studies, listed in section 10.4.2.2, Table 10-2.*
Workshops	*Use by professionals in training workshops, listed in section 10.4.2.3, Table 10-3. There are three programs to train professionals at workshops, each with their own usefulness to train professionals in the use of MoST and its KB:* • Demonstration (2h)*: only presentations, demonstrations and discussion.* • Short workshop (4-6h)*: Introductions, demonstrations and hands-on experiences for MoST and its KB, leaving out the more complex setting with multi-user and multi-domain modelling projects.* • Long workshop (12-16h)*: Introductions, demonstrations, hands-on experiences for MoST and its KB and role-playing, including the more complex setting with multi-user and multi-domain modelling projects.* *All workshops so far were of the* demonstration *or* short workshop *type.*
Courses	*The training program used for students so far, is a variant to the* short workshop program *adapted for modelling courses at Wageningen University (see section 10.4.2.4, Table 10-4). Students were trained for 4 hours to become familiar with MoST and its modelling KB and worked subsequently for 60 hours on model-based problem solving cases.*
Reviews[6]	*The KB and MoST were intensively reviewed by the following external reviewers, not involved in the HarmoniQuA project:* *1. Pasky Pascual (Environmental Protection Agency, Washington, USA);* *2. Nils Ferrand (Cemagref, Montpellier, France);* *3. Hugh Middlemis (Aquaterra, Kent Town, Australia).*

[5] The test plan has been reported in Rocha (2002) and the results of three test series in Rocha (2003, 2004, 2005).

Reuse[7]	*Reuse experiments, in which the developed technology is re-used for another process. The technology consists of the more abstract layers of the process ontology and the tool MoST.*
User questionnaire[8]	*On-line questionnaire for users of the training material with general and detailed questions. The training material has been designed to train students and professionals in how to use MoST and its knowledge base. A summary of the results can be found in section 10.4.2.5, Table 10-5. The full questionnaire can be found in Appendix-I.*
Professional survey[9]	*A survey with paper questionnaires asking for the opinion of the professional community on quality assurance in modelling and in particular on HarmoniQuA's knowledge base and MoST. The professional community consists of those people who are directly involved in applying models and those managing modelling studies (with modelling experience).* *Out of the 985 questionnaires that were sent only 105 were completed and returned (response rate of approximately 11%).*
Stakeholder survey[10]	*A survey with paper questionnaires asking for the opinion of stakeholders on quality assurance in modelling and in particular on HarmoniQuA's knowledge base and MoST. Stakeholders include all persons not directly involved in modelling, e.g. water managers, interest groups (agricultural/industrial associations and green NGOs), planners, policy makers and concerned members of the public.* *Out of the almost 577 questionnaires 108 were completed and returned (response rate of approximately 19%).*

Scientific output	*Reviewed scientific output, such as:*
	• *Refereed Journal papers*:
	    o Henriksen et al. (200x);
	    o Refsgaard and Henriksen (2004);
	    o Refsgaard *et al*. (2005);
	    o Scholten *et al*. (2007).
	• *Refereed Conference papers*:
	    o Bergfeld (2005);
	    o Blind *et al*. (2004);
	    o Blind *et al*. (2005);
	    o Kassahun, *et al*. (2004);
	    o Kassahun and Scholten (2006);
	    o Old *et al*. (2005);
	    o Olsson *et al*. (2004);
	    o Refsgaard *et al*. (2006a, b);
	    o Scholten *et al*. (2004);
	    o Scholten and Kassahun (2006);
	    o Scholten *et al*. (2006).
	• *Refereed Book Chapters*:
	    o Scholten and Beulens (2006).
	An overview can be found on www.harmoniqua.org/public/Product/papers.htm.

Details of some of the summarized tests are given in the sections 10.4.2.2 – 10.4.2.5. The other tests are described elsewhere and references are given in Table 10-1.

---

[6] The reviews of Pascual and Ferrand were presented at the HarmoniQuA Full Meeting '*Evaluation of the full version of MoST*' in Lisbon, 14-17 October 2004; PowerPoint slides are available but not publicly. The third review is published by Middlemis (2004).
[7] This is briefly discussed in Scholten *et al*., 2006.
[8] The user questionnaire can be found on http://informatics.wur.nl/most-evaluation/questions.asp.
[9] Reported in Old and Packman (2005).
[10] Reported in Old *et al*. (2005).

10.4.2.2 Testing in case studies

MoST and its modelling knowledge base for water management have been developed in three phases, each having their own version of the product:

1. May 2003: (incomplete) *prototype version* providing basic guidance and monitoring;
2. June 2004: *full version* with most of the intended functionality;
3. December 2005: *final version* with all planned functionality, except for the advisory component, that aimed at learning from previous modelling projects on similar types of problems to solve.

After each of the first two development phases, a series of ten test cases have been carried out, of which the results were used in de the next development phase to fix bugs, to improve MoST and its KB and finally to provide for a proof of principle (Table 10-2).

Model-based problem solving studies have often aspects belonging to more than one (water management) domain. Therefore the case studies in HarmoniQuA can be categorized in three groups: single domain case studies, multi-domain case studies and integrated case studies. The latter type consists of multi-domain case studies, of which one of the domains is socio-economics. The first series of 10 case studies was aimed at including single domain case studies and multi-domain case studies. The second series of 11 case studies aimed at multi-domain and integrated case studies.

Table 10-2. HarmoniQuA case studies form the first (2003-2004) and second (2004-2005) test series. There are 3 types of case studies: (1) single domain, (2) multi-domain, and (3) integrated (i.e. multi0domain including socio-economics). Legends: FF=flood forecasting; PR=precipitation-rainfall; HD=hydrodynamics (including sediment / morphology); GW=groundwater; WQ=water quality; BI=biota; SE=socio-economics.

Case #	*Partner*	*Case study*
First series of case studies (end 2003- begin 2004)		
1	Cemagref [11]	PR + FF + SE (aspects)
2	BfG[12]	HD + SE (aspects)
3	BfG	HD + WQ + BI + SE (aspects)
4	SMHI[13]	PR + WQ + SE (aspects)
5	Vituki[14]	HD + WQ + GW
6	LNEC[15]	FF + HD (incl. morph) + SE (aspects)
7	DHI-cz[16]	PR + HD + SE
8	WL | DH[17]	HD (including morphology)
9	WL | DH	HD + WQ + BI
10	NTUA[18]	PR + WQ
Second series of case studies (end 2004 – begin 2005)		
11	Cemagref	PR + FF + (S)E-aspects
12	BfG	HD + PR
13	BfG	GW + BI
14	BfG	HD (including changes of water level, flood forecasting, etc.)
15	SMHI	PR +WQ +(S)E-aspects
16	Vituki	HD + WQ
17	LNEC	FF + HD (including sediment) + (S)E-aspects
17	DHI-cz	PR + WQ + FF
18	WL | DH	HD + WQ
19	WL | DH	PR + FF
20	NTUA	PR + WQ
21	CEH[19]	PR (mostly application of single domain model codes)

[11] Cemagref, Centre National du Machinisme Agricole, du Génie Rural, des Eaux et Forêts, Groupe Hydrologie, U.R. Qualité et Fonctionnement Hydrologique des Systemes Aquatiques, Antony cedex, France
[12] BfG, Bundesanstalt für Gewässerkunde, Koblenz, Germany
[13] SMHI, Swedish Meteorological and Hydrological Institute, Norrköping, Sweden
[14] VITUKI Plc, Budapest, Hungary
[15] LNEC, Laboratório Nacional de Engenharia Civil , Lisbon, Portugal
[16] DHI Hydroinform a.s., Department of Water Resources and River Hydraulics, Prague, Czech Republik
[17] WL | Delft Hydraulics, Delft, Netherlands
[18] NTUA, National Technical University of Athens, Department of Civil Engineering, Athens, Greece

10.4.2.3 Testing in workshops

In the period 2004-2006 15 National and International workshops have been organized by all HarmoniQuA partners focusing on facilitating the adoption of MoST by potential users. An overview of these workshops is presented in Table 10-3.

Table 10-3. List of HarmoniQuA Workshops.

Date	Workshop	Workshop type
18 November 2004	Gareth Old (CEH), Copenhagen, Denmark	International, short workshop[20]
4-8 May 2005	Jan Spatka (DHI-cz), Baile Felix, Rumania	National, demonstration workshop[21]
1 June 2005	Anker Højberg (GEUS), Copenhagen, Denmark	National, short workshop
6 July 2005	Gabor Balint (Vituki), Nyiregyhaze, Hungary	National, demonstration workshop
28 September 2005	Jonas Olsson (SMHI), Norrköping, Sweden	National, demonstration workshop
27 October 2005	Simon Groot (WL I DH), Huub Scholten (WU), Delft, Netherlands	National, short workshop
8 November 2005	Jens Christian Refsgaard (GEUS), Helsinki, Finland	National, short workshop
10 November 2005	Gareth Old (CEH), Wallingford, UK	National, short workshop
29 November 2005	Jan Spatka (DHI-cz), Prague, Czech Republic	National, demonstration workshop
2 December 2005	Charles Perrin (Cemagref), Antony, France	National, short workshop
16 December 2005	Gareth Old (CEH), Melbourne, Australia	National, short workshop
16 December 2005	Maria Mimikou, Maria Kapetenaki, Christina Panagiotopoulou (NTUA), Athens, Greece	National, demonstration workshop
25 January 2006	João Rocha (LNEC), Lisbon, Portugal	National, short workshop
6 April 2006	Ingo Heinz (Uni-Do), Huub Scholten (WU), Osnabrück, Germany	National, demonstration workshop
6-8 September	Jan Spatka (DHI-cz), Bologna, Italy	National, demonstration workshop

10.4.2.4 Testing in courses

The training material of HarmoniQuA's MoST has been used in an unknown number of courses at several universities. Only a few are used here in the testing procedure. These include only courses at Wageningen University at MSc-level, because no results of other courses were available.

Table 10-4. List of courses in which MoST and its KB have been and are used at Wageningen University.

ID	Description
INF-20306[22]	Elementary Programming and Modelling
INF-30806[23]	Advanced Modelling and Simulation
INF-31806[24]	Models for Forest and Nature Conservation
GRS-30306[25]	Spatio-Temporal Modelling

[19] Centre for Ecology and Hydrology, Wallingford, Hydrological Risks Division, Wallingford, UK
[20] A *short workshop* takes 0.5-1.0 day and consists of presentations, hands-on experience and discussions.
[21] A *demonstration workshop* takes 2 hours and consists of presentations, demonstrations and discussions.
[22] https://csa.wur.nl/wpage8/xpage.aspx?xml=vak_xml.iread?Vak:Vak%20id=91UOQPLGAYQ9O2MM$Ondeenheid:Gidsjaar=2005$Oplsrt:Oplsrt=R&xsl=/bois/xsl/vak.xsl&css=gids.css&lang=usa&app=bois
[23] https://csa.wur.nl/wpage8/xpage.aspx?xml=vak_xml.iread?Vak:Vak%20id=920K564VRSFUZ2IX$Ondeenheid:Gidsjaar=2006$Oplsrt:Oplsrt=R&xsl=/bois/xsl/vak.xsl&css=gids.css&lang=usa&app=bois
[24] https://csa.wur.nl/wpage8/xpage.aspx?xml=vak_xml.iread?Vak:Vak%20id=920JZ3NA40UBY70L$Ondeenheid:Gidsjaar=2006$Oplsrt:Oplsrt=R&xsl=/bois/xsl/vak.xsl&css=gids.css&lang=usa&app=bois

10.4.2.5 User questionnaire

Since 2004 an online questionnaire enables users of MoST and its KB to evaluate and comment on the product. The questionnaire has several sections. The section with *general information* of the user consists of questions on operating software, type courseware used, user type (student, scientist, professional modeller, etc.), years of experience and why he/she is interested in using MoST. An overall impression section is presented in Table 10-5. Furthermore, there are other sections evaluating *details on the training material* and *details on MoST* and its modelling guidelines. The full questionnaire is presented in Appendix-I.

Table 10-5. Summary of the overall impression of users of MoST and its KB, based user questionnaire. The full questionnaire and its results can be found in Appendix-I.

Overall impression (n=50)	*%*
Impression training material	
Training web material is useful introduction to MoST.	84
The training web material requires a HarmoniQuA trainer	46
Training was of sufficient duration	78
Impression MoST	
I would like to use MoST in my modelling work.	48
I believe the guidance it offers will be useful.	76
I believe the monitoring functionality it contains will be useful.	62
I believe the reporting functionality it contains will be useful.	86
I believe MoST will enhance the quality of modelling work	74

Some overall findings will be discussed here. Users think that the training material is useful and of a sufficient duration, while half of them believe a demonstrator/trainer is not necessary to do the training. Half of the respondents like to use MoST in their modelling work. Three quarters find the guidance from MoST's KB useful, two thirds consider the monitoring functionality as very useful, 86% think the reporting functionality is useful and 74% believes that MoST will enhance the quality of their modelling work.

Some overall conclusions of the more detailed parts of the questionnaire will be presented here. Respondents evaluated most issues with a mark between 7 and 8 (on a scale of 1-10). The following items got a lower mark. The user interface got a 6.7 (probably due to the fact that many respondents used a preliminary version of the software[26]). The guidance on modelling (quality, quantity and clarity) was also appreciated with marks between 6 and 7, probably because it was rather difficult for students, but it was without serious errors (mark 7.8). The glossary was satisfactory, but not very useful (mark 6.8). The monitoring part of MoST was also appreciated with reasonable marks, but it appeared difficult to understand.

10.4.3 Test criteria

Just like in software engineering, quality is a rather complex issue in knowledge engineering and knowledge bases. Quality can be measured or evaluated only in a subjective way. Software quality is usually measured by evaluating various *quality factors* (McCall *et al*., 1977, Boehm, 1978). McCall *et al*., (1977) distinguish three phases in software development, each with its own quality factors:

- **Product operation phase**:
    - *Correctness*: the extent to which a program satisfies its specification and fulfils the customer's mission objectives.
    - *Reliability*: The extent to which a program can be expected to perform its intended function with required precision.
    - *Efficiency*: The amount of computing resources and code required by a program to perform its function.

[25]https://csa.wur.nl/wpage8/xpage.aspx?xml=vak_xml.iread?Vak:Vak%20id=920JXKF6CSKLD4QO$Ondeenheid:Gidsjaar=2006$Oplsrt:Oplsrt=R&xsl=/bois/xsl/vak.xsl&css=gids.css&lang=usa&app=bois
[26] The final version 3.1.5 was released in March 2006.

- *Integrity*: The extent to which access to software or data by unauthorized persons can be controlled.
  - *Usability*: The effort required to learn, operate, prepare input, and interpret output of a program.
- **Product revision phase**:
  - *Maintainability*: The effort required to locate and fix an error in a program.
  - *Flexibility*: The effort required to modify an operational program.
  - *Testability*: The effort required to test a program to ensure that it performs its intended function.
- **Product transition phase**:
  - *Portability*: The effort required to transfer the program from one hardware and/or software system to another.
  - *Reusability*: The extent to which a program (or parts of a program) can be reused in other applications – related to the packaging and scope of the functions that the program performs.
  - *Interoperability*: The effort required to couple one system to another.

Boehm and colleagues (Boehm *et al*., 1978) extended McCall's quality factors. A more accepted system is provided by the International Standardization Organisation (ISO 9000-3, 1991), which follows to some extent a similar approach with quality factors, criteria and metrics. To test the total ontological framework proposed in this book a more practical approach has been followed. Not a static standard is used, but a new set of criteria is developed which should be evaluated with the set of tests of section 10.4.2. Here the criteria will be discussed, based on a preliminary set of criteria, defined by Scholten and Beulens (2005).

As outlined before, testing of knowledge-based systems is difficult, as it includes many aspects. In Table 10-6 the following *knowledge-based system components* have been evaluated, each with its own criteria:

1. KB structure;
2. KB process decomposition;
3. KB content;
4. MoST;
5. Training material and help.

Because of the complexity and the diversity of the aspects to test a list of the criteria used is given in Table 10-6.

Next to the list presented in Table 10-6, other criteria can be defined, e.g. the need for an ontological approach or the effectiveness of the dissemination effort. These are left out as they do not fit in the test approach followed here.

Table 10-6. Evaluation criteria for the *modelling ontology, MoST* and the *training material*. This list is adapted from Scholten and Beulens (2005).

#	Test criteria	Explanation
1 KB structure		
1a	Correctness	*Does it capture the intuitions of domain experts?*
1b	Completeness	*Can everything that is needed be represented?*
1c	Consistency	*Is it a correct ontology?*
1d	Granularity	*Not too detailed, not too abstract, but fitting the knowledge level required to do the job*
1e	Flexibility	*Does the ontological approach allow changing the structure at reasonable cost?*
2 KB process decomposition		
2a	Correctness	*Do the decomposition and the flowchart capture intuitions of domain experts and their mental model of modelling processes?*
2b	Completeness	*No gaps (steps, tasks, activities, methods)?*
2c	Redundancy	*No unintended duplications in steps, tasks, activities, methods?*
2d	Consistency	*Does the decomposition contain contradictions?*

2e	Transparency	*Do the flowchart and the rest of the decomposition give a transparent view on the structure of the process?*
2f	Granularity	*Not too detailed, not too abstract, but fitting the knowledge level required to do the job*
3 KB content		
3a	Correctness	*Capturing intuitions of domain experts?*
3b	Completeness	*No gaps (steps, tasks, activities, methods)?*
3c	Redundancy	*No unintended synonyms?*
3d	Consistency	*Consistency in handling concepts? Contradictions included?*
3e	Meaningfulness	*Can intended users understand it?*
3f	Necessity	*The need to have KB content on modelling, for water management or in general.*
3g	Acceptance	*The (content of the) KB should be accepted by a substantial part of the modelling community[27].*
4 MoST		
4a	Correctness	*Does MoST function correctly, i.e. without errors and software bugs?*
4b	Reliability	*Is the system available, functioning, and accessible by a project team when needed?*
4c	Functionality	*Does MoST function according to the requirements?*
4d	Adequacy	*Does MoST adequately support the daily practice of professionals?*
4e	Learnability	*Is MoST appropriate for teaching novice model users, including students?*
4f	Necessity	*The need to have a tool to support the work of a multidisciplinary team for model-based water management.*
4g	Acceptance	*MoST should be accepted by a substantial part of the modelling community.*
5 Training material + Help		
5a	Correctness	*Does the training material (website) function correctly, i.e. without errors and bugs?*
5b	Usefulness professionals	*The training material is useful for professionals, if a professional can use it to learn how to use MoST and its KB.*
5c	Usefulness students	*The training material is useful for students, if a student can use it to learn how to use MoST and its KB.*
6 KB technology		
6a	Reusability KB structure	*Is the ontological structure of the KB reusable?*
6b	Reusability MoST	*Is the MoST tool useful for other processes?*

10.4.4 Test results

The combination of the test types of section 10.4.2 (Table 10-1) have been used to evaluate the criteria (section 10.4.3, Table 10-6) selected for each knowledge-based system component. The results are summarized in and all details are in Appendix H (Table H-1).

[27] The modelling community includes researchers and (professional) practitioners in modelling for water management.

Table 10-7. Summary of test results of the *modelling ontology* using the tests of Table 10-1 and the evaluation criteria of Table 10-6 for each knowledge-based system component. Details are presented in Appendix H (Table H-1).

Test results
1 KB structure
Not many comments on the ontological structure of the KB have been received. Ontologies are hard to understand for domain experts without knowledge engineering experience. Useful indications on how to change the ontology emerged indirectly when putting the pieces of modelling knowledge into instances of the ontology.
2 KB process decomposition
Several times we received requests for changing the structure of the modelling process. These requests came from project partners and from the 'wider modelling society'. All remarks were carefully evaluated and several have been used to improve the decomposition of the modelling process in ***tasks*** *and of* ***tasks*** *into* ***activities****. These changes were mainly related to the order of the tasks, their dependencies, but also to the decomposition in tasks and the activities associated with the tasks. Implementing these changes was quite easy and not time consuming, because of the flexibility provided by the ontological approach. Typically, substantial changes required a few hours to a single day of work to incorporate the changes in MoST and its KB.*
3 KB content
Feedbacks on the content of the decomposition elements included long lists of errors, wishes and comments. But all respondents so far appreciated the guidance provided by the KB and found it useful, especially for novice users of MoST.
4 MoST
The results for criterion 'Adequately supporting daily practice of professionals' accumulated in the first test series were promising and directed the redesign of MoST to a more powerful level. These tests led to a long series of small suggestions that have been discussed and partially implemented. The first test series also identified more important shortcomings. The modelling support provided by MoST was insufficient in two aspects. The version used for these tests was too much focused on single users and on monodisciplinary projects. Extra functionality for modelling teams and for multidisciplinary projects has been implemented in the full version of MoST. The second test series with the full version of MoST provided other needs for change. Applying MoST and its KB in university courses resulted in a similar request, i.e. for multi-user support and multidisciplinary application domains. *Many testers (professionals) and students wanted MoST to enable them to work at a higher level: not only fulfilling tasks by doing activities, but also detailing what team members do at a task level only. In the latter case the activities are just headings in the model journal contribution for that task.*
5 Training material + Help[28]
Testing the training material led to many relatively small changes. Many of the national workshops for professionals are scheduled for this year, so answers on criterion 'usefulness professional' are not available yet. But using the training material in student courses ('usefulness students') showed very promising results. Students learn very quickly (a few hours) how to use MoST in a training case study and apply it in their problem oriented education projects, in which small groups of students have to solve environmental problems with a model. The guidance provided by the KB directed them effectively through the network of tasks, of which modelling projects usually consist. This approach also proved to be more efficient than the textbook approach on Good Modelling Practices used in the same courses in the past.
6 KB technology
Using the HarmoniQuA technology in AquaStress showed that technology related criteria 'Is the KB ontological structure reusable?' and 'Is the tool MoST useful for other processes?' are fulfilled so far, although successfully using the technology in AquaStress and other processes in future will enhance confidence in reusability.

10.5 Testing the problem and object system ontology

The next part of the ontological framework is the *problem and object system ontology*, discussed in Chapter 8 (see also Figure 10-1, central branch). The development of this branch of the ontological

[28] The training material and the help-system were not yet available at the time of the reviews.

framework is an indirect result of developing a series of models e.g. ecosystem models and bivalve ecophysiology models. These models aim at representing and simulating relevant facets of reality, including structural and behavioral aspects. The knowledge for these models is derived from scientific research and organized in a separate ontology: the *problem and object system ontology*. Chapter 8 discusses this ontology and refers to the underlying body of expertise derived from literature.

Despite its extensive description in Chapter 8 and Appendix F, the process to build a knowledge base, discussed in Chapter 3 requires four elements, i.e. the development of a proper *ontological structure*, a *knowledge acquisition*, *storage* of acquired knowledge in instances of the ontology and *software application(s)* that can use to knowledge base. This branch of the ontological framework consists of an ontological structure with ontological layers of increasing levels of detail and populated with knowledge bivalve ecophysiology, but without a tool that actually can use the KB. The latter makes testing rather fuzzy and indirect. Testing is therefore limited to three test types: *face evaluation*, *application* and *reuse*. These tests are defined in Table 10-8 and subsequently applied on the *ontological structure* (i.e. how the various concepts are related) and *ontology content* (i.e. the semantics of concepts) of the *problem and object system ontology*. The test results are summarized in Table 10-9.

Table 10-8. Test types used to test the *problem and object system ontology*.

Name	Explanation
Face evaluation	*Inspection by a limited set of experts of the domain factual knowledge, i.e. generic problem knowledge (*ontological layer 1*), specialized problem knowledge (*ontological layer 2*) and problem knowledge for a specific application domain (*ontological layer 3*).*
Application	*Applying the ontology when using factual* problem and object system *knowledge to the model ontology, i.e. pieces of factual knowledge from the* problem and object system ontology *should fit into the proper instances of the* model ontology.
Reuse	*Reuse experiments, in which the developed* problem and object system ontology *is re-used for another domain or discipline.*

The criteria used to evaluate the *problem and object system ontology* are similar to those used to test the modelling ontology (see Table 10-6). The test results are summarized in Table 10-11 and detailed results are given in Appendix H (Table H-2).

Table 10-9. Summary of tests of the *problem and object system ontology*. Details are presented in Appendix H (Table H-2).

Test results
1 Problem and object system ontology structure
Testing the problem and object system ontology structure *was difficult, as the best form of testing is using the ontology in applications and by tools. The few tests performed for this ontology did not reveal many shortcomings and errors. Despite the modest level of testing, the development of such an ontology has value in itself.*
2 Problem and object system ontology content
The problem and object system ontology *was applied to develop a series of models on bivalve ecophysiology. From applying the* content *of the* problem and object system ontology*, it can be concluded that the knowledge organized in this way is useful to develop simulation models. Therefore this leaf of the ontological framework can be assumed as validated, although in a very limited sense.*

The development of a tool[29] that can use the ontology (structure and content) to support model building for a class of models would need a more comprehensive set of tests. Such tool has not been developed so far and it would require substantial resources to develop it. Therefore it is left out here.

10.6 Testing the model ontology

The *model ontology* is the final part of the proposed ontological framework for model-based water management (see also Figure 10-1). The *model ontology* has been developed for a wide class of models, but the models discussed in Chapter 9 all belong to the class of *simulation models*. Similar to the *problem and*

[29] Such a tool will be briefly discussed in Chapter 11, section 11.10.

object system ontology, the *model ontology* is hard to evaluate. No tool[28] has been developed for this ontology, which hinders its testing.

Two aspects of the model ontology have been tested: the *structure*, i.e. how the various concepts are related, and the *content*, i.e. the semantics of concepts. These aspects of the model ontology have been tested with 3 types of tests, summarized in Table 10-10.

Table 10-10. Test types used to test the *model ontology*.

Name	Explanation
Face evaluation	*Inspection by a limited set of experts on simulation models*
Application	*Applying the ontology when building (simulation) models*
Reuse	*Reuse experiments, in which the developed* model ontology *is re-used for another domain or discipline.*
Publishing	*Publishing scientific papers in peer reviewed journals guarantees an objective view on (at least) the model resulting from the ontology.*

The criteria used to evaluate the *model ontology* aspects are similar to those used to test the modelling ontology (see Table 10-6). The test results are summarized in Table 10-11 and detailed results are given in Appendix H (Table H-3).

Table 10-11. Tested aspects of the *model ontology* with the results per criterion and per test type used.

Test results
1 Model ontology structure
Testing the model ontology structure *was difficult, as the best form of testing is applying an ontology in applications and by tools that use the ontology. Despite the modest level of testing, the development of such an ontology has value in itself.*
2 Model ontology content
Testing the model ontology content *was also difficult, as no tools have been developed that use the model ontology content. As soon as such tools have been developed more shortcomings will probably be detected.*

All results from these tests have been used to improve the model ontology.

10.7 Conclusion

The four parts of the ontological framework (*meta-ontology*, *process ontology*, *problem and object system ontology* and *model ontology*) are tested with different intensities, due to the respective levels of maturity of the ontologies. The *process ontology* and its associated tool, *MoST*, has been developed in the HarmoniQuA project and large parts (the process technology, i.e. the ontological structure and the tool MoST) are used in the AquaStress project. Therefore the *process ontology* and its support tool, *MoST*, are more mature and have more thoroughly been tested. Testing is a continuous process, leading to improvements in all aspects of the ontological framework, the associated tools and the technology. Until now tests have been encouraging and continue to support the idea that our approach is a successful one.

All tests used to verify and validate the ontological framework, belong to the category ‘soft testing’, i.e. tests in which persons are involved instead of automatic, formal testing by software tools. The results are therefore rather inexact and will not completely answer how well an aspect matches with a criterion.

Testing of the four ontologies does not provide any formal proof. Each test, successfully passed, will enhance confidence in the tested (part) of the ontology for the evaluated criteria. Although many tests have been applied, large parts of the ontological framework proposed here have not been sufficiently tested or not tested at all.

The *meta-ontology* was easiest to test, as it was developed based on direct needs for terminology and does not contain concepts and the associated relations that belong to one of the other three ontologies: the *processes ontology* (instantiated for the modelling process), the *problem and object system ontology* (instantiated for bivalve ecophysiological knowledge) and the *model ontology* (instantiated for bivalve ecophysiological models).

The *process ontology* and more precisely the modelling ontology and its modelling support tool MoST have been tested in a comprehensive way. From the tests many shortcomings have been discovered, which could be repaired during the development of the ontology, its instances and the tool. The overall result can therefore be characterized as a rather mature and well-tested part of the ontological framework, which can be further improved for modelling by extending its modelling domains. The ontological technology (MoST and the more generic ontological layers of the process ontology) will be adapted to support projects in other processes.

The *problem and object system ontology* was most difficult to test, as it contains domain knowledge, organized in *ontological layers* of increasing specialization. The most detailed parts of this ontology contain expert knowledge in very specialized and isolated areas of science (this ontology has been instantiated for bivalve ecophysiology). Face evaluation by some of experts is a valuable contribution to testing the content of this ontology, but experts are not necessarily good judges of the more generic parts of this ontology. This impedes a straightforward test approach. Large parts of the ontology describing and structuring bivalve ecophysiology was published in peer reviewed journal papers and therefore accepted as a result of sound scientific research. Therefore, it can be assumed that the content of this ontology is acceptable, as far as it matches with the published knowledge. The ontology structure is even less well tested. Using the more generic parts for other problem domains may reveal further shortcomings.

The *model ontology*, the fourth part of the proposed ontological framework, has been tested best as far as it has been used to build models. These models aimed at summarizing bivalve ecophysiological processes that rule physiological adaptation to changing ecological conditions for bivalves, as far as these are not genetic adaptations. These models have been published in refereed journals and have been reviewed by others, e.g. Beadman *et al*. (2002). Therefore large parts of the proposed model ontology, in particular the more detailed and specialized parts, are sufficiently tested.

In conclusion, it can be stated that the ontological framework is useful but not tested comprehensively in all its aspects and parts. Especially the value of combining the four parts of the ontological framework has not seriously been explored. A combination of the ontological framework elements may enable a more integrated support of modelling. Such an approach should use a modelling ontology to support the defining of the problem and object system and to develop models that can help solve the essential parts of the problem at hand. This integrated approach requires a complex tool that supports model-based studies in all its aspects. It should ease defining problem/object systems, assist teams in modelling and help in specifying models. Furthermore, such tool should facilitate grasping how problems, modelling and models interact. Problems and associated object systems have representation demands that determine – to some extent – the choice of proper modelling paradigms, which in turn have representation and solution power, delineating what can be represented of the real world in the problem/object system ontology. The proposed ontological framework can best be seen as part of an infrastructure for such a tool.

10.8 References

Beadman, H.A., R.I. Willows and M.J. Kaiser, 2002. Potential applications of mussel modelling. Helgoland Marine Research 56, 76-85.

Bench-Capon, T., D. Castelli, F. Coenen, L. Devendeville-Brisoux, B. Eaglestone, N. Fiddian, A. Gray, A. Ligeza and A. Vermesa, 1999. Report on the 1st International Workshop on Validation, Verification and Integrity Issues of Expert and Database Systems. Information Research 4. http://informationr.net/ir/4-3/paper55.html.

Bergfeld, T. (2005): Das EU-Projekt HarmoniQuA: Qualitätssicherung zur Erhöhung der Zuverlässigkeit bei der Modellierung von Flusseinzugsgebieten. In: Nacken, H., Bartusseck, S. & Sewilam H. (eds) Entscheidungsunterstützung in der Wasserwirtschaft – von der Theorie zum Anwendungsfall. Forum für Hydrologie und Wasserbewirtschaftung 10.05, 223-227.

Blind, M.W., R. Moore and H. Scholten, 2004, Scientific opportunities created by HarmonIT, HarmoniRiB and HarmoniQuA, Proceedings of Watermatex 2004, 6th International Symposium on Systems Analysis & Integration Assessment, 3-5 November 2004, Beijing, China,19-27.

Blind, M.W., R.V. Moore, H. Scholten, J.C. Refsgaard, S.A. Borgvang, C. Giupponi, I. Borowski, M. Estrela, D.G. George, J. Froebich, I. Zsuffa, P. Vanrolleghem and W. De Lange, 2005, Current Results Of The EC-sponsored Catchment Modelling (CatchMod) Cluster. In: A. Zerger, R.M. Argent (Eds.),

MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, ISBN: 0-9758400-2-9, 1210-1216, December 2005.
Boehm, B.W., Brown, J.R., Kaspar, H., Lipow, M., MacLeod, E.J., & Merritt, M.J. (1978). Characteristics of Software Quality. New York: North-Holland.
Dadkhah, C. and A. Abdollahzadeh Barfouroush, 2004, OAV-VVT Expert, An active system for Verification and Validation of Knowledge Base Based on Ex-OAV KB, ICSE & INCOSE 2004 CONFERENCE, Las Vegas, Nevada, USA, Sep 15-18, 2004, http://jedai.afia-france.org/repository/15.pdf#search=%22Dadkhah%20validation%22.
Gómez-Pérez, A., 2004. Ontology Evaluation. In: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, vol., Springer, ISBN, pp. 251-274.
Henriksen, H.J., J.C. Refsgaard, A.L. Højberg, N. Ferrand, P.J.A. Gijsbers and H. Scholten, 200x (resubmitted). Public participation in relation to quality assurance of water resources modelling (HarmoniQuA). Environmental Modelling & Software.
IEEE, 1998. IEEE standard for software verification and validation, IEEE 1012-1998, 75 pp.
ISO 9000-3 (1991). Quality management and quality assurance standards - part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software. Geneve: International Organization for Standardization.
Kassahun, A. and H. Scholten, 2006, A knowledge base system for multidisciplinary model-based water management, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006.
Kassahun, A., H. Scholten, G. Zompanakis and C. Gavardinas, 2004. Support for model based water management with the HarmoniQuA toolbox. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Complexity and Integrated Resources Management, Transactions of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs, Manno, Switzerland, ISBN 88-900787-1-5, pp. 1282-1287.
Kramer, M.R. and H. Scholten, 2001, The Smart approach to modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering., Delft, The Netherlands, June 26-29, 2001, TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0.
McCall, J.A., Richards, P.K., & Walters, G.F. (1977). Factors in software quality. Springfield, Va.: National Techical Information Service.
Middlemis, H., 2004. Benchmarking best practice for groundwater flow modelling, The Winston Churchill Memorial Trust of Australia, Canberra, Australia, Fellow Report, 45 pp. http://www.churchilltrust.com.au/res/File/Fellow_Reports/Middlemis%20Hugh%202004.pdf.
Old, G.H. and J.C. Packman (Eds.), 2005. Position paper on professional response. Centre for Ecology and Hydrology, Wallingford, HarmoniQuA-report D-WP4-2, 35 pp. http://harmoniqua.wau.nl/public/Reports/Response/Final%20Stk%20Rpt%2012Apr05.pdf.
Old, G.H., J.C. Packman and H. Scholten, 2005, Supporting the European Water Framework Directive: The HarmoniQuA Modelling Support Tool (MoST). In: A. Zerger, R.M. Argent (Eds.), MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, ISBN: 0-9758400-2-9, 2825-2831, December 2005.
Old, G.H., M. Robinson and J.C. Packman (Eds.), 2005. Position paper on stakeholder response. Centre for Ecology and Hydrology, Wallingford, HarmoniQuA-report D-WP4-3, 20 pp. http://harmoniqua.wau.nl/public/Reports/Response/Final%20Stk%20Rpt%2012Apr05.pdf.
Olsson, J., Scholten, H., Arheimer, B., Andersson, L., 2004. Quality assurance support tool for catchment-based modelling: a test on the HBV-NP model for eutrophication assessment, 8th International Conference on Diffuse/Nonpoint Pollution, 24-29 October 2004. International Water Association, Kyoto.
Oreskes, N., K. Shrader-Frenchette and K. Belitz, 1994. Verification, Validation,and Confirmation of Numerical Models in the Earth Sciences. Science 263, 641-646.
Popper, K.R., 1959. The logic of scientific discovery. Unwin Hyman Ltd, London, 2nd, 480 pp.
Preece, A., 2001. Evaluating Verification and Validation Methods in Knowledge Engineering. In: R. Roy (Ed.), Micro-Level Knowledge Management, Morgan-Kaufman, San Francisco, pp. 123-145.

Refsgaard, J. C. and H. J. Henriksen. 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27: 71-82.
Refsgaard, J. C., H. J. Henriksen, B. Harrar, H. Scholten, and A. Kassahun. 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software, 20: 1201–1215.
Refsgaard, J.C., A.L. Højberg, H.J. Henriksen, H. Scholten, A. Kassahun, J.C. Packman and G.H. Old, 2006a, Quality assurance support through most (HarmoniQuA), Keynote speech, Modflow and More, Golden, Colorado, May 22-24, 2006.
Refsgaard, J.C., A.L. Højberg, H.J. Henriksen, H. Scholten, A. Kassahun, J.C. Packman and G.H. Old, 2006b, Quality Assurance of the modelling process, The XXIV Nordic Hydrological Conference, Vingsted Centret, Denmark.
Rocha, J.S. (Ed.) 2003. Test Report of HarmoniQuA's Modelling Support Tool: Version 1.0. Laboratório Nacional de Engenharia Civil, Lisbon, HarmoniQuA-report D-WP2-5a, 29 pp.
Rocha, J.S. (Ed.) 2004. Test Report of HarmoniQuA's Modelling Support Tool: Version 2.0. Laboratório Nacional de Engenharia Civil, Lisbon, HarmoniQuA-report D-WP2-5b, 29 pp.
Rocha, J.S. (Ed.) 2005. Test Report of HarmoniQuA's Modelling Support Tool: Version 3.0. Laboratório Nacional de Engenharia Civil, Lisbon, HarmoniQuA-report D-WP2-5c, 28 pp.
Rocha, J.S., 2002. Testing the products of WP-2. In: H. Scholten, S.A. Osinga (Eds.), Requirement analysis report of the HarmoniQuA knowledge base and tools, Wageningen University, Wageningen, pp. 89-91.
Sargent, R.G., 1984. Simulation model validation. In: T.I. Oren, B.P. Zeigler, M.S. Elzas (Eds.), Simulation and model-based methodologies: an integrative view, NATO ASI Series F: Computer and Systems Sciences, Springer Verlag, Berlin, etc., pp. 537-555.
Scholten, H. and A. J. M. Beulens (2005). Testing ontological support for multidisciplinary model-based problem solving for water management. Knowledge Creation and Integration for Solving Complex Problems. The 19th International Workshop on Complex Systems Modeling (CSM) jointly with the 6th International Symposium on Knowledge and Systems Sciences (KSS), August 29-31, 2005. M. Makowski. Laxenburg, Austria, IIASA: 51-55.
Scholten, H. and A. Kassahun, 2006, Supporting multidisciplinary model-based water management projects: a user perspective, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006.
Scholten, H. and A.J.M. Beulens, 2006. Managing mathematical modelling by guiding and monitoring. In: A.S. Kazi, P. Wolf (Eds.), Real-Life Knowledge Management: Lessons from the Fields. KnowledgeBoard (www.knowledgeboard.com) in cooperation with VTT - Technical Research Centre Finland (www.vtt.fi), ISBN 952-5004-72-4, pp. 213-232, http://www.knowledgeboard.com/knowledgebank/book.html.
Scholten, H. and M.W.M. Van der Tol, 1994a. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.
Scholten, H. and M.W.M. Van der Tol, 1994b. Towards a metrics for simulation model validation. In: J. Grasman, G. Van Straten (Eds.), Predictability and nonlinear modelling in natural sciences and economics, Kluwer Academic Publishers, Dordrecht, pp. 398-410.
Scholten, H. and M.W.M. Van der Tol, 1998. Quantitative validation of deterministic models: when is a model acceptable? In: M.S. Obaidat, F. Davoli, D. DeMarinis (Eds.), The proceedings of the Summer Computer Simulation Conference, SCS, The Society for Computer Simulation International, San Diego, CA,USA, pp. 404-409.
Scholten, H., A. Kassahun and A.J.M. Beulens, 2006a, Use and reuse of an ontological knowledge base framework. In: M. Makowski (Ed.), The 20th International Workshop on Complex Systems Modeling (CSM), Laxenburg, Austria, 28-30 August, 2006, IIASA, 45-48.
Scholten, H., A. Kassahun and J.C. Refsgaard, 2006b, Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006, Research Publishing, Volume 3, 2231-2238.
Scholten, H., J.C. Refsgaard and A. Kassahun, 2004. Structuring multidisciplinary knowledge for model based water management: the HarmoniQuA approach. In: C. Pahl, S. Schmidt, T. Jakeman (Eds.), Complexity and Integrated Resources Management, Transactions of the 2nd Biennial Meeting of the

International Environmental Modelling and Software Society, iEMSs, Manno, Switzerland, ISBN 88-900787-1-5, pp. 1288-1293.

Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.

Scholten, H., M.W.M. Van der Tol and A.C. Smaal, 1998, Models or measurements? ANNUAL SCIENCE CONFERENCE, Cascais, Portugal, International Council for the Exploration of the Sea.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J. Noort, 1999. Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice. STOWA/RWS-RIZA, Utrecht/Lelystad, ISBN 90-5773-056-1, 149 pp.

11 Discussion

“Begin at the beginning and go on till you come to the end; then stop.”
(Lewis Carroll, pseudonym of Charles Lutwidge Dodgson, mathematician and logician, 1832-1898)

Content of Chapter 11

11.1 Problems to be solved

Managers of problem solving projects in water management have to arrive at a shared vision on the nature and extent of a modelling project, in which solutions have to be found to a stated management problem. Such a vision entails the scope of the study, the solution approach, expected results, duration, costs and resources used. Thereafter, for a commissioned project, the problem is to execute the study within its specifications including quality assurance issues. In this way, transparency is guaranteed and projects are easier to audit and reconstruct (Scholten *et al*., 2006, 2007).

Strict quality assurance requirements for modelling projects are caused and fuelled by a multitude of problems and bad experiences with model based studies in the past. Refsgaard *et al*. (2005) and Scholten *et al*. (2007) give several reasons for these problems, including ambiguous terminology, a lack of mutual understanding between key-players, bad practice with regard to input data, inadequate model set-up, insufficient calibration/validation, model use outside of its scope, insufficient knowledge of some processes, miscommunication between the modeler and the end-user, overselling of model capabilities, confusion on how to use model results in decision making and a lack of documentation and transparency of the modelling process.

An additional complicating factor is related to the changing character of model-based problem solving projects from monodisciplinary, single person and academic oriented research model studies into multidisciplinary, decision support oriented projects, in which teams consisting of members with different background and different roles have to cooperate to complete the complex job. Modelling in multidisciplinary teams enables exploring more complex questions, at the same time this makes cooperation in teams more difficult. Team members with different scientific backgrounds encounter more communication problems, which makes managing multidisciplinary model-based water management projects a cumbersome affair (Scholten *et al*., 2006).

Nowadays there are often legal prerequisites to decision making that require public participation in the decision making process, e.g. the Water Framework Directive and similar legislation. Any approach to lower hurdles in model-based problem solving should consider this aspect.

Next to this *process dimension* of multidisciplinary modelling, a consistent, well-structured view is needed on the problem to be solved and on the models, which are instrumental in model-based problem solving. The difficulties, associated with the *problem dimension* and the *model dimension*, are similar to those related with the process itself.

The ontological framework proposed in this book is only detailed for an incomplete number of branches, which limits its usefulness, but the proposed ontological structure is open for extensions, i.e. knowledge on other processes, problems, models and model types.

11.2 A framework for multidisciplinary model-based problem solving

In this book a framework has been proposed that aims to support multidisciplinary teams in model-based problem solving, focused on continuous simulation models, applied to environmental and ecological applications in water management. The framework consists of a structure of four (related) ontologies, covering (1) *modelling*, (2) *problems and object systems*, (3) *models* and (4) the *meta-ontology* with basic terminology needed to describe the other tree ontologies.

The four ontologies consist of a *structural part* with concepts and relations between the concepts and of *instances*, in which the (structural) concepts are 'filled-in'. The structural part of the ontological framework resembles a tree with many branches. Some of these branches are fully described, while others are only suggested without details. The mainstream branches shown in front of Figure 10.1 (Chapter 10) are complete and expanded in detail. The left main branch (*process ontology*) is worked out for modelling using (mainly) continuous simulation models and resulted in a knowledge base (KB) with modelling guidelines for water management. As side branch the water stress mitigation process is defined within the context of the AquaStress project. This is hardly discussed in this book. The central branch (*problem and object system ontology*) and the right branch (*model ontology*) are populated with detailed knowledge on bivalve ecophysiology and associated simulation models. These populated branches are described in Chapters 5 (*meta-ontology*), Chapter 6 (*process/modelling ontology*), Chapter 8 (*problem and object system ontology*) and Chapter 9 (*model ontology*).

One of the innovative qualities of the proposed framework is the layered structure of the ontologies, from generic to specialized and detailed (see Figure 10-1). This not only provides an appropriate level for specialists to describe all their special details, but also facilitates communication on the included knowledge between less specialized team members at a less detailed level.

Next to the ontologies (and associated KBs) a set of tools were developed and /or used. The free, open source ontology editor and knowledge base framework Protégé has been used to set-up of the structured ontologies with increasing levels of specialization and the KB with modelling guidelines. *Protégé* has been extended by building a plug-in for exporting XML, according to a predefined XML-format, interpretable for the modelling support tool that has to co-operate with the KB. To fill the KB a (web based) KB-Editor has been developed, which acts as front-end between domain experts, unskilled in knowledge engineering, and the knowledge base implemented in Protégé. The KB Editor also handles authorization issues and assures that users can work concurrently in the system by locking knowledge elements under revision to overcome authors entering conflicting updates (Kassahun and Scholten, 2006). This KB-Editor is now extended to better support other types of collaborative research processes. Furthermore, the KB-editor enables the development of glossaries related to the content of a process KB, including the modelling KB.

To support teams in multidisciplinary projects a Modelling Support Tool, MoST, has been developed, initially focused on modelling, but now extended to the Process Support Tool, ProST, which aims to support other collaborative research processes (Scholten *et al*., 2006, 2007). The tool is discussed in Chapter 7.

11.3 Scientific merit

Compared to state-of-the-art Quality Assurance guidelines for model-based water management, e.g. the Dutch GMP Handbook (Van Waveren *et al*., 1999), the Bay-Delta modeling protocol (BDMF, 2000) and the Australian groundwater modeling guidelines (Middlemis, 2000), the modelling guidelines, discussed in Chapter 6, are more complete and flexible (Scholten *et al*., 2007). This is reflected in its knowledge base (KB) by including several modeling application domains, targeting different types of users and serving various levels of job complexity. Furthermore, and most important, the ideas and implementation of the supporting tool MoST is novel, as no other guideline group has attempted to prepare such a tool (Scholten *et al*., 2007).

Regarding the *process ontology*, instantiated for model-based problem solving in water management, and the other ontologies (*meta-ontology*, *problem and object system ontology* and *model ontology*), the most innovative aspect of the proposed ontological framework consists of its structure, which is articulated in two dimensions. In the first dimension three main ontological branches are distinguished, i.e. *process ontology*, *problem and object system ontology*, and *model ontology*. In the second dimension each of the three main branches have ontological layers of increasing specialization (Figure 10-1). This ontological structure is new and facilitates conversation on its content between human users of the ontology and computer programs. Human communication about the content of the proposed ontologies is realized in the following way: an ontological *layer x* provides the language and terminology to discuss concepts and relations at the level of the more specialized (and thus less generic) *layer x+1*.

Finally, the layered ontology structure enables the reuse of at least the more generic (less specialized) layers for other purposes. This reuse is only realized and therefore tested for the *process ontology* and not for the other ontologies (see Chapter 10).

11.4 An ontological approach

A major objective of the present book was to develop a framework to support collaborative (research) processes in general and more specifically multidisciplinary model-based problem solving, especially for water management, but also for other environmental and ecological problems. The support consists of knowledge bases (KBs) and tools. The KBs have ontological structures and the KB content is stored in instances of the ontological concepts.

The *process ontology* is the most mature ontology and its associated KB is the result of substantial efforts within the HarmoniQuA project (Scholten *et al*., 2007). This modelling KB passed all stages of an ontological knowledge base development with (1) the design of an ontological structure, (2) the development of a knowledge acquisition tool, (3) the implementation of the KB as instances of the ontological structure and (4) the building of a tool to use the KB for some purpose. Publications on ontologies (Borst, 1997, Chandrasekaran *et al*. 1998, 1999, Gruber, 1993, 1995 and many others) focus on knowledge engineering aspects of ontologies and on developing domain factual knowledge ontologies. For our purpose an ontological approach was only instrumental in the development of a modelling KB for water management. The requirements set to this KB (explicit modelling knowledge, flexible structure, shared by many experts, easy to update and maintain, properly secured, platform independent) and its main goals (providing guidance and straightforward use of the modelling knowledge in Modelling Support Tool, MoST) directed the design of the KB onto an ontological track. This design has been initially reviewed internally, i.e. within the HarmoniQuA project and subsequently shared with the larger modelling community in water management by publications (Refsgaard *et al*., 2005, Scholten *et al*., 2007). Results so far (using the KB in an educational context, in 21 professional test cases and some other projects) showed that this was a sound design decision, as most requirements have been met so far. Substantial efforts will be spent to handle gaps and inconsistencies in the content of the KB and to make it a generally accepted methodology for modelling in water management. The publications on the resulting modelling guidelines have been appreciated by the scientific modelling community[1].

The other ontologies, i.e. the *problem ontology* and the *model ontology* are less mature and they have been developed on a more *ad hoc* basis. The *problem ontology* can best be seen as a side-product of the series of papers on bivalve ecophysiology models (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005). This ontology was based on the models used in these papers (see also Chapter 8). In contrast to the process ontology for modelling, which has been developed and populated by a group varying between 5 and 25 persons, the knowledge of the *problem ontology* has been derived from the expertise of only a few researchers during a long series of discussions on developing and subsequently structuring simulation models representing the physiological response of ‘blue mussels’, *Mytilus edulis* L. on varying ecological inputs, including food supply. Its structure, specified in the more detailed and specialized ontological layers, has been largely evoked from building this series of models. During model development natural ontological layers became visible as well as a kind of decomposition of the body of knowledge at hand. These two have been used to set-up the *problem ontology*. Reusing the knowledge for upgraded versions of the model made it more consistent and developing a model for a different, but closely related bivalve species, i.e. ‘cockles’, *Cerastoderma edule* (L.), made it more complete (Rueda *et al.*, 2005).

11.5 Knowledge bases

11.5.1 Modelling methodology

A major aim of the HarmoniQuA project was to establish a multidisciplinary modelling methodology for water management. This objective has been reached by realizing a detailed knowledge base with modelling expertise for multidisciplinary modelling for water management, providing users in different roles guidance on what they have to do, within all supported domains and for different job complexities. Furthermore, the Modelling Support Tool (MoST), discussed in Chapter 7, also supports the work of modelling teams in their daily practice by monitoring what modelling teams do and facilitating project management by providing detailed data on what is done by whom in the project and on resources spent (Scholten *et al*., 2007).

The part of the methodology on how to model for water management, is not new, as the content of the KB is based on existing knowledge and existing guidelines for model-based water management (Scholten, 1999, 2001, Van Waveren *et al*., 1999, Scholten *et al*., 2000, 2001, Blind *et al*., 2000, Middlemis, 2000, BDMF, 2000). But the existing methodology has been substantially extended by (1) making it explicit in the publicly

[1] The paper *Refsgaard, J. C., H. J. Henriksen, B. Harrar, H. Scholten, and A. Kassahun. 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software, 20: 1201–1215* is awarded as one of two runners-up in the ‘Best 2005 research paper in EMS’. It was selected by the panel of Associate Editors on the grounds of quality and relevance. The impact factor of this journal in 2005 was 1.351 (Journal Citation Reports® 2005, published by Thomson Scientific).

accessible KB, (2) developing MoST to support actual modelling by multidisciplinary teams playing different roles (modeler, water manager, auditor, stakeholder, concerned members of the public) in modelling projects for water management. This support consists of monitoring all activities, methods used, project management data and helping to produce reports for various audiences. Furthermore a glossary for model-based water management has been developed with about 1000 terms. In this way the resulting methodology is more powerful than the existing ones and – in this sense – innovative. Nevertheless, it fits seamlessly in the daily practice of professionals in model-based water management (Scholten *et al*., 2007).

The success of the developed methodology (i.e. providing modelling guidance, monitoring what team members do in a shared model journal and reporting model journal content) stands or falls on the basis of two conditions: (1) large parts of the professional community of modelers for water management should use it and (2) there should be an achieved consensus on the content of the knowledge base, i.e. an agreement on Best Modelling Practices. These two additional objectives are not yet met and substantial efforts have to be invested to realize these requirements.

11.5.2 KB for modelling and other collaborative processes

Making modelling knowledge explicit and representing it in an ontological form are the major benefits of MoST's knowledge base (KB). The ontological approach makes improving, changing and updating the KB easy. This is an essential prerequisite for ensuring that the KB can become a body of knowledge shared by large parts of the professional modelling community. Its specificity for model-based water management does not hinder extending it to other disciplines or application domains, thanks to the flexibility provided by its ontological set-up. This partly demonstrated by using parts of the process ontology for an other application domain (water stress mitigation in the AquaStress project; see Chapter 6 and Chapter 10)This benefit of the approach followed leads to one of its disadvantages at the same time. Because the KB can simply be adapted, it will appear as somehow arbitrary to parts of the target group. The present state of the KB can be seen as a cross section in time of relevant and sufficient modelling knowledge. It is an intermediate product open for improvements and updates, if scientific progress requires this (Scholten *et al*., 2007).

11.5.3 KB for problems / object systems

The *problem and object system ontology* is less mature than the *modelling ontology*, as it lacks a proper implementation in Protégé and there are no tools to use its content directly. Nevertheless it can be useful, as skeleton with consistent terminology to describe knowledge on problems and associated (aspects) of object systems, especially as this knowledge has to be represented into a (simulation) model. The *problem and object system ontology*, proposed in Chapter 8, has subsequently been tested in the set of models on bivalve ecophysiology, described in Chapter 9 (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005). It can be concluded that knowledge organized in this way is instrumental in the development of simulation models and facilitates reusing part of the *problem and object system ontology*. The usefulness of this ontology would substantially be increased with a tool, which can structure such body of factual domain knowledge about problems and object systems and pass it on to integrated modelling software, as is briefly discussed in Chapter 10 (section 10.7) and in this Chapter (section 11.10).

11.5.4 KB for models

The *model ontology* is also rather immature, as it lacks a proper implementation and tools to use it. Nevertheless it appears useful in the limited number of models that were based on it. To improve the *model ontology*, it should be integrated in model development tools and – somehow – use the *problem ontology* content and translate that directly or with help of a modeler into a model.

The *model ontology* is based on the expertise and (simulation) traditions of the Information Technology group at Wageningen University, which has 30 years of experience in modelling & simulation (M&S) methodology and M&S software (many publications including Elzas, 1988, Scholten *et al*., 1990, Klepper *et al*., 1994, Scholten and Van der Tol, 1994, Kramer and Scholten, 2001). The *model ontology* has further been molded by the requirements set by the development of implemented and published models.

Using the *problem ontology* and the *model ontology* in the series of papers in peer reviewed journals shared it with a wider expert community, led to acceptance with some enthusiasm of our view on the knowledge systems for modeling[2].

11.6 MoST and ProST

The Modelling Support Tool, MoST aims at supporting the full life-cycle for multidisciplinary teams in model-based problem solving for water management (see Chapter 7). The support consists of the following features: *providing guidance* (from the *modelling ontology* or some other *process ontology*), *monitoring* (and storing what multidisciplinary teams do in a model project plus their results in model journals[3]) and *reporting* (the content of a model journal).

In the initial design phase of MoST, professional modelers had a vague picture of the final product. The requirements for a modelling support tool in water management proved to be very complex and most persons involved or interested were not fully convinced of its feasibility. The present state of the Modelling Support Tool, MoST, proves that the comprehensive expertise of multidisciplinary modelling teams can be represented in and supported by a software tool that really helps them in their daily practice. Its professional users do not regard MoST as a straitjacket hindering these experts to model according to one's own habits and institutional practices. MoST provides modelling guidance, which is filtered for user type, water management domain, job complexity and the modelling task at hand. MoST also invites modelling team members to keep records of what they do, integrates the work of all team members and stores these records in an ontologically structured model journal. Furthermore, MoST keeps track of project management data and allows managers to check modelling project progress. Finally, MoST filters the information in model journals to reports adapted for different audiences, including modelers, managers, auditors, stakeholders and interested members of the public (Scholten *et al*., 2007).

The experiences with MoST so far are promising and MoST has been used in 21 test case studies within the HarmoniQuA project and several modelling projects thereafter. An evaluation of MoST can be found in Chapter 10 and based on these test results it may be concluded that MoST is a novel way of supporting multidisciplinary modelling for water management (Scholten *et al*., 2007).

Usually, support for modelling (simulation models oriented) consists of environments to set-up and work with simulation models. At present some tools support coupling of (sub)models into a sound, composite model to solve a specific problem, e.g. Harmon-IT's OpenMI[4] (Blind *et al*., 2005, Gijsbers and Gregersen, 2005). Other tools support only parts of the modelling process. I am not aware of any tool supporting the whole modelling lifecycle as MoST does. Therefore a comparison of MoST with other tools is not appropriate as this would be restricted to comparing functionalities instead of determining how effective the modelling support is (Scholten *et al*., 2007).

Despite the focus on modelling for water management, the tool can also be used for other modelling disciplines and even for other collaborative research projects. Therefore, the tool will be transformed to the more generic Process Support Tool (ProST) and extended with the following new features: defining new *user types* and *domains* (stored in the modelling KB and used by ProST), *multilanguage support* of ProST (GUI, message, etc.), in a redesign of the tool all functionalities will be organized more consistently in components to allow and facilitate extensions of the tool and its flexibility, guidance on how to set-up the whole system (tool, KB, project server, multilanguage issues) and some support will be provided to initialize and maintain multilanguage guidelines (whether for modelling or other collaborative processes, all based on the same ontological structure in the more generic ontological layers as is discussed in Chapter 6).

---

[2] Scholten and Smaal were the proud winners of the Dresscher Prize 1998-1999 for their paper: *Scholten, H. and Smaal, A.C., 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.*

[3] Because not only the modelling process is supported, but also other processes the term *project journal* can be used instead of *model journal*.

[4] More information on OpenMI at: www.openmi.org.

11.7 Use and reuse of the framework

The ontologies of the framework have been used in different ways and with different intensities. The *meta-ontology* has been used by all others and was extended, as soon as terminology to develop the other ontologies was missing. The *problem ontology* and *model ontology* can be used to describe problems and associated object systems in case a (simulation) model has to be used to solve a problem at hand. All instances of these two ontologies are related to bivalve ecophysiology problems and models (see Chapter 8 and Chapter 9).

The *modelling ontology*, a more specialized instantiation of the *process ontology,* has been used intensively and always in combination with MoST (at present ProST). The discussion on the use of the framework will therefore focus on this combination. This use can be seen from two angles: a user perspective and a technological perspective (Scholten and Kassahun, 2006).

In the user perspective several types of user can be distinguished, including problem owner (water manager), modeller, auditor, stakeholder and public. These user types (roles) use the combination of ontological KB for modelling and MoST (ProST) in a different way. The *problem owner / manager* uses it mainly in the tendering phase of a project, where the KB content acts as template of what has to be done in a project, and further to check the progress of a project. *Modellers* use it mainly to keep notes of what they do and with which results. Moreover, they can use model journals to communicate their modelling activities to their colleague modellers and the manager. *Auditors* use the model journal to see what has been done and furthermore they use the scoreboards to evaluate the work of the modelling team. *Stakeholders* (with a direct stake in the problem at hand) and interested members of the *public* (including interest groups and private persons with interest) have to be informed and in the case of water management, the present water framework directive (WFD) requires that they are consulted too. Preferably (but under no obligation) they participate in a more active sense (co-designing, co-decision making and co-implementating). Using MoST (ProST) and the modelling KB enables these two types of users to be informed and to react to what happens in the project, as they are consulted in so called review tasks (Scholten and Kassahun, 2006). Finally, MoST (ProST) helps the professional community involved in multidisciplinary model-based water management by focusing on a state-of-the-art-modelling methodology, which is the content of the modelling KB. The modelling methodology in this KB is open for comments within the limits of the KB editing authorizations. This openness stimulates a broad discussion and facilitates achieving agreement within a large group (Scholten and Kassahun, 2006).

The appreciation of MoST and its KB has emerge from 21 case studies, 6 university courses and 15 professional workshops, a professional survey, a stakeholder survey and a user questionnaire, as is presented and discussed in Chapter 10. From a user perspective this appreciation can be summarized as follows. The *(water) managers* will not use MoST themselves, but let some project manager play MoST's manager role. In their first projects, *modellers* feel MoST as a straitjacket that forces them to work according to the guidelines: just 'overhead' instead of 'help'. Later they experience the ease of use and the benefits of making explicit what they actually did. *Auditors* perceive the use of MoST as a prerequisite that facilitates their review work. *Stakeholders* and *public* find MoST difficult, but if an expert mediator guides them, they can appreciate it as a tool for real participation in modelling projects (Scholten and Kassahun, 2006).

Ontological layer 0 (meta-ontology), *ontological layer 1* (generic process knowledge), *ontological layer 2* (generic modelling knowledge) of the modelling ontology and MoST are defined as the *modelling support technology*. In HarmoniQuA we combined this modelling support technology with ontological layer 3 (knowledge for model-based water management) and referred to this combination as *support for multidisciplinary model-based water management*. In addition to the generic modelling knowledge, seven water management domains are supported at present (hydrodynamics, groundwater, precipitation-runoff, flood forecasting, surface water quality, biota and socio-economics). There are plans to extend this set of domains with 'activated sludge modelling'. For this purpose ontological layer 3 has to be extended with knowledge about this 'new' domain. MoST and its KB are also used in complex, model-based water management projects in Sweden, Denmark, UK, Netherlands and Germany (Scholten and Kassahun, 2006).

The *modelling support technology* can also be reused for other types of (simulation) modelling, e.g. environmental modelling, crop growth modelling (e.g. in SEAMLESS, www.seamless-ip.org). This will require a new content of *ontological layer 4*, at present containing knowledge for model-based water management (Scholten and Kassahun, 2006).

If other processes than modelling have to be supported, the *modelling support technology* has to be reduced to a *process support technology* by leaving out ontological layer 2 (modelling knowledge) and, obviously, layer 3 (knowledge for model-based water management). The *process support technology* then consists of ontological layers 0 (meta-ontology), layer 1 (generic process knowledge) and ProST, i.e. the more generic version of MoST. Subsequently, new ontological layers 2 and 3 have to be developed containing structured knowledge about the new process. An example of using the *process support technology* for other processes can be found in the AquaStress project[5]. This project aims at water stress mitigation by providing various water stress mitigation options (technical, management, institutional and others), scientific evaluation of options (multi-criteria analysis, simulation, case based reasoning, etc.) for case studies at specific sites and by supporting participatory processes, in which stakeholders and public participate in selecting and evaluating water stress solutions (Scholten and Kassahun, 2006). Other examples of reusing parts of the technology include a recent initiative in Denmark to develop a KB for geological modelling, the implementation process of WFD (Water Framework Directive) and supply chain management. The latter two have not been implemented.

In conclusion, it can be said that MoST and its KB are seen as useful by intended users, although they use it in different ways. The ontological layered structure allows reusing parts of the *modelling ontology*, whether for modelling (other domains of simulation modelling, other modelling paradigms[6]) or for other collaborative processes.

Software tools should not require a steep training curve, as this does not encourage their use. User manuals are tedious and appear not to be very efficient at resolving obstacles encountered by novice users. Therefore, a multimedia approach has been chosen, consisting of presentations, screen recording movies, exercises on using MoST in a test case modelling project and written background material. All training material and a help-system are available through a website (www.harmoniqua.org/training and www.harmoniqua.org/training/help). This facilitates using the training material in a course setting, but it also allows novice users to work individually. The multimedia set-up appeared to be appealing and helped novice users (students and professionals) to become skilled at using MoST substantially faster than with manuals only. This claim is based on subjective observations during the courses and workshops, in which the training material was used. The effectiveness and efficiency of the training material will be evaluated later (Scholten *et al*., 2007).

The *problem ontology* and the *model ontology* have only be used in the development of the series of models on bivalve ecophysiology and – although its use has had a positive effect on the modelling studies (Smaal and Scholten, 1997, Scholten and Smaal, 1998, 1999, Rueda *et al.*, 2005) – its use for other applications cannot be evaluated here. These ontologies have to be integrated in modelling software or tools that use the ontologies before their reuse can be evaluated.

11.8 The framework in a QA perspective

In comparing what has been achieved by realizing MoST and its modelling KB for water management with the five stages of the Simulation Maturity Model (SMM) proposed by Scholten and Udink ten Cate (1999), one can conclude that SMM's third stage of maturity (*defined*) has been achieved by developing the modelling KB. The development of the tool part of MoST facilitates modelling teams to manage the process. This achievement brings modelling to SMM's fourth stage (*managed*), at least modelling for water management. Although this statement is speculative, one can expect that the advisory component of MoST will appear to be a first step to the final stage of SMM (*optimized*) by learning from previous experiences and improving present modelling (Scholten *et al*., 2007).

[5] See www.aquastress.net.
[6] *Model paradigm* has been defined in Appendix G (Table G-1) as: Model type and associated format, solver and methodology.

11.9 Problems solved

The problems in model-based water management, presented in Chapter 1 (section 1.4), are partly solved with the development of MoST and its KB. Scholten *et al*. (2007) discuss how MoST and its KB try to deal with these difficulties will be discussed here. The glossary is an answer to *ambiguous terminology used in modelling*, while the explicit guidelines in the KB help to promote *mutual understanding in multidisciplinary modelling teams*. *Bad practice* cannot be completely banned in modelling projects, but the guidelines direct professionals to model according to 'Good Modelling Practices' and to avoid some of the unprofessional practices, including *careless handling of input data*, *inadequate model set-up*, *insufficient calibration and validation and model use outside of its scope*. The widespread problem of *too few data* or *the poor quality of available data* will obviously not be solved by the methodology and tools presented here;, this is also the case for *insufficient knowledge on processes hindering ecological (biota) modelling*. *Miscommunication between the modeler and the end-user on the possibilities and limitations of the modelling project* and *overselling of model capabilities* are still possible, but modelers are guided to avoid these pitfalls. Furthermore, the model journals enable end-users to do or order proper audits and to check the translation of model results into end-user advice. This will also help to avoid *confusion on how to use model results in decision-making*. Furthermore, the model journals are a direct answer to the problem of *lack of documentation and transparency of the modelling process, leading to projects, which hardly can be audited or reconstructed*. Finally, MoST and its KB distinguish socio-economics as a separate and essential domain of water management, which helps to avoid *insufficient consideration of economic, institutional and political issues and a lack of integrated modelling*.

In addition to these problems solved for modelling in water management, the proposed framework also aims to solve similar problems for modelling in other domains. This aim has not been worked out yet and needs future research.

This book focuses on facilitating and improving the process of multidisciplinary (simulation) modelling with MoST and the modelling knowledge base. But when trying to improve modelling, one has to include the problems to be solved and their object systems in the proposed ontological framework. Moreover, this book identifies a need for a more formal, ontology based, approach to describe models, directly linked to the object system ontology for the problem at hand. Finally, a meta-ontology is proposed for the basic terminology. This framework is more than just MoST and the modelling ontology, but the modelling tools and knowledge base are more extensively realized. The other elements of this framework are not less important, but their usefulness is less general, because of their limited use so far and the limited convincing tests passed as was outlined in Chapter 10. Using the problem (plus OS) ontology of Chapter 8 and the *model ontology* of Chapter 9 for practical purposes proved their value.

The full power of the proposed framework cannot be proved to be effective, unless it is implemented and supported by a toolbox that supports its use. This last aspect is outside the scope of this book and will require a further substantial effort. Despite the limited realization of the framework, it is reasonable to claim that MoST (ProST) and its KB improve the quality of model-based problem solving, especially for water management, by providing guidance and making the modelling process more transparent.

11.10 Research questions

In Chapter 1 the following research questions have been outlined:

1. How can a *problem* and its associated *object system* be described in a way that enables the selection, development or instantiation of an appropriate model to solve the problem at hand?
2. How should a *model* be described in a way that makes it scientifically sound and instrumental to solve the problem at hand?
3. How should *modelling* be done according to good modelling practices and in a transparent way, allowing reconstruction model-based problem solving projects?

To what extent are these research questions adequately answered? The third of these research questions has most intensively been taken care of and is discussed in this book. Next to a *process ontology*, instantiated for modelling, tools have been developed in the HarmoniQuA project. The KB editor helps to set up process ontologies and their instances to establish and maintain knowledge bases. MoST, a Modelling Support Tool, and its successor, ProST, the Process Support Tool, developed for the project AquaStress, support multidisciplinary teams in execution their explicitly defined process in projects. Furthermore, the modelling ontology and the tools have been adequately tested (see Chapter 11).

The other two research questions (1. and 2.) have been answered, but these answers are only partly implemented in knowledge bases and software tools that help using them. Despite the limited realization of these two parts of the proposed framework, some of its usefulness has been shown in Chapter 8 (research question 1.) and Chapter 9 (research question 2.). The structured knowledge on problems (including their associated object systems) and on models appeared to be constructive. This has been shown in Chapter 8, where some problems on bivalve ecophysiology have been structured ontologically and in Chapter 9, where some ontology based models have been discussed in relation to solving the structured problems of Chapter 8.

If the research questions are sufficiently answered, a new – more generic question – can be formulated as follows. *Does the proposed ontological framework improve the quality of modelling?*[7] The new generic question can be answered positively, as the proposed ontological framework helps to improve modelling and makes it improves the chances of reconstructing what has been done. But many side products, e.g. the branches of the three main ontologies (*modelling*, *problem and object system*, *model*) are not fully developed and discussed, but only briefly indicated. Moreover, the proposed ontological framework has a promising power, which is not fully revealed yet. Therefore an integrated tool should be developed that uses more of the proposed ontological framework. Such a tool should improve support of problem formulation, should help defining a relevant object system for that problem, on which a model (or a set of alternative models) can be based and should facilitate the other modelling tasks. MoST's model journals should accordingly contain instances of relevant models and descriptions of problem and object system. The complexity of such an extended tool positions it beyond the scope of this book and realization will require substantial effort and resources. But the modelling community, consisting of researchers and professionals, seems to be waiting for such a promising extension of the modelling paradigm that facilitates working with alternative models and supports a better, more extensive analysis of the problem and its model-based solutions. The terms *representation demand* and *representation power*, introduced by Beulens and Scholten (2001) and briefly discussed in Chapter 1 can be managed in a better way within the proposed ontological framework and its realization in an integrated tool for modelling, although discrepancies between representation demand and representation power will not automatically be solved with such an integrated tool.

The next question to be answered here is about the effectiveness and efficiency of developing a proprietary ontology and process description format and not using some existing one, e.g. GLIF (GuideLine Interchange Format). This item is introduced in Chapter 2 and discussed in Chapter 6. In an appropriate selection procedure, several formats would have been tested and the best would have been selected. Such a procedure was not followed. Developing one's own format facilitates fulfilling many requirements. The set-up proposed here with a framework with four ontologies, which in turn are organized in layers of increasing specialization, provides a format able to represent a complex structure with more generic parts in the top layer and more detailed ones in the specialized layers. This layered structure of the ontologies in the framework provides a richer representation power for knowledge on processes, problems and models than other formats do by allocating concepts at the right level of specialization.

In Chapter 1 a gap between model-based problem solving for policy or decision making has been discussed. This gap is widely perceived as negative and growing. This book fills parts of this gap by making modelling a more transparent process, which is open for critical assessment and allows reviewers / auditors to reconstruct what multidisciplinary modelling teams have done.

11.11 Multidisciplinary cooperation

Chapter 4 aims at identifying differences between mono– and multidisciplinary model-based problem solving. Chapters 5–10, subsequently attempt to overcome perceived shortcomings in multidisciplinary processes. The solutions consist of the proposed ontological framework and especially the *process ontology*, instantiated for modelling and the tool MoST that supports cooperation in multidisciplinary model-based problem solving. In this way the proposed solutions contribute to a theoretical base for multidisciplinary

[7] The new question is an extension of the basic question of chapter 1: *how to improve the quality of modelling in order to increase its credibility?*

model-based problem solving by providing vocabulary and a methodology to extend the monodisciplinary paradigm, as proposed by Kuhn (1970), to a multidisciplinary paradigm. MoST and its more flexible successor, ProST that aims at supporting a wider class of cooperation projects, facilitates cooperation between researchers, professionals, stakeholders and interested members of the public in complex projects. This approach also supports *public participation*. Participatory processes with public participation in complex decision making can have three levels of involvement: (1) being informed, (2) being consulted and (3) active involvement, i.e. discussions, influence on the policy agenda, participatory design of solutions, involvement in decision making and participating in implementation (Pahl-Wostl, 2002, 2005, Pahl-Wostl and Hare, 2004, Ridder *et al.*, 2005). MoST facilitates informing and consulting stakeholders and public. If members of these groups are added to the problem solving team, they can read or write in parts of the model journal within the limits of their authorization. The third level of participation is more active. To facilitate the latter, most intensive, type of participatory involvement, MoST's KB with guidelines can easily be adapted.

11.12 A final remark

In this book, I did not want to propose a framework, a formal system that is just *true* or *false*. The proposed framework is only a way of structuring things for a purpose, a meta-description of what I want to describe when using models to solve some problem at hand. In my view, ontologies do not aim at developing a 'construct' describing 'all there is' in the sense of Wittgenstein's *Tractatus Logico-Philosophicus* (Wittgenstein, 1921[8]). Wittgenstein's first proposition claims that there is nothing more than facts[9]. Wittgenstein's *facts* refer to things, objects and their relations as expressed in language. What not is expressed in facts does not exist. In this way Wittgenstein's *facts* are comparable with Zeigler's *base model*[10] (Zeigler, 1976), although Zeigler's *base model* has a more limited connotations restricted to models and modelling. The proposed ontological framework aims only to describe and structure relevant knowledge (about processes, modelling, models, problems and object systems). Its relevance is determined by its purpose. I do not aim at describing the *world*, but only aspects that are relevant for some purpose. These aspects are views on relevant concepts of *what is*, filtered by science, scientific observation, technology and some purposeful reason. Therefore, there can be no single, *perfect* ontology, but many useful ones depending on different purposes. This book aimed to provide an ontological framework, useful for its purpose, i.e. better modelling practices. If one invests substantial resources for such a construct, it is better to make it as open as possible, i.e. reusable for other problems, models and modelling paradigms[11]. Finally, reusing the proposed framework will show its genuine value.

11.13 References

BDMF, Protocols for Water and Environmental Modelling. Bay-Delta Modelling Forum. http://cwemf.org/Pubs/Protocols2000-01.pdf, 2000.

Beulens, A. and H. Scholten, 2001. Model solving software and modeling paradigms: a challenge for developers. In: M. Makowski (Ed.), Extended abstract of 15th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, August 27-29, 2001, vol., IIASA, Laxenburg, Austria, ISBN, pp. 4-8.

Blind, M.W., H. Scholten and R.H. van Waveren, 2000. Model for success. Water21, October 2000.

---

[8] See http://www.kfs.org/~jonathan/witt/tlph.html.

[9] Wittgenstein's first proposition in his major work the claims:

1. The world is everything that is the case.
   - 1.1. The world is the totality of facts, not of things.
     - 1.11. The world is determined by the facts, and by these begin all the facts.
     - 1.12. For the totality of facts determines both what is the case, and also all that is not the case.
     - 1.13. The facts in logical space are the world.
   - 1.2. The world divides into facts.
     - 1.21. Any one can either be the case or not be the case, and everything else remains the same.

[10] In chapter 9 a *base model* is defined as: … a model capable of accounting for all input-output behavior of the real system. In other words, it is valid in all the allowable experimental frames. In any realistic modelling and simulation area, the base model can never be fully known, although certain aspects of its description may be expected as known. Since the base model provides a complete explanation of the behavior of a real system, it may be expected to comprise many, many components and interactions.

[11] *Model paradigm* is defined in chapter 9 as: Model type and associated format, solver and methodology.

Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. University of Twente, The Netherlands, Enschede.
Chandrasekaran, B., J. Josephson, and V. Benjamins. 1998. Ontology of tasks and methods. in Eleventh Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada.
Chandrasekaran, B., J. Josephson, V. Benjamins, 1999. What are ontologies and why do we need them? IEEE Intelligent Systems 14, 20-26.
Elzas, M.S., 1988. Desktop modelling and simulation environments. In: R.C. Huntsinger, W.J. Karplus, E.J. Kerckhoffs, G.C. Vansteenkiste (Eds.), Simulation environments: symbol and number processing on multi and array processors, Society for Computer Simulation International, San Diego, pp. 389-393.
Gijsbers, P.J.A. and J.B. Gregersen, 2005, OpenMI: A glue for model integration. In: A. Zerger, R.M. Argent (Eds.), MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, December 2005, Modelling and Simulation Society of Australia and New Zealand, 648-654, http://www.mssanz.org.au/modsim05/papers/gijsbers.pdf.
Gruber, T. R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5((2)): 199-220.
Gruber, T. R., 1995. Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43:907-928.
Kassahun, A. and H. Scholten, 2006. A knowledge base system for multidisciplinary model-based water management, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006.
Klepper, O., M.W.M. Van der Tol, H. Scholten and P.M.J. Herman, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.
Kramer, M.R. and H. Scholten, 2001, The Smart approach to modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering., Delft, The Netherlands, June 26-29, 2001, TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0.
Kuhn, T., 1970. The structure of scientific revolutions. International Encyclopedia of Unified Science, University of Chicago Press, Chicago, 2nd, 210 pp.
Middlemis, H., 2000. Murray-Darling Basin Commission. Groundwater flow modelling guideline. Aquaterra Consulting Pty Ltd., South Perth. Western Australia.
Pahl-Wostl, C., 2002. Towards sustainability in the water sector - the importance of human actors and processes of social learning. Aquatic Sciences 64, 394-411.
Pahl-Wostl, C., 2005. Actor based analysis and modeling approaches. The Integrated Assessment Journal 5, 97-118. http://journals.sfu.ca/int_assess/index.php/iaj/article/view/167/110.
Pahl-Wostl, C. and M. Hare, 2004. Processes of social learning in integrated resources management. Journal of Community & Applied Social Psychology 14, 193–206.
Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.
Ridder, D., E. Mostert and H.A. Wolters (Eds.), 2005. Learning together to manage together: improving participation in water management. University of Osnabrück, Osnabrück, ISBN 3-00-016970-9, 115 pp., http://www.harmonicop.info/handbook.php
Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (Cerastoderma edule L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.
Scholten, H., 1999. Good Modelling Practice. In: M. Makowski (Ed.), Abstracts of the 13th JISR-IIASA Workshop on methodologies and tools for complex system modeling and integrated policy assessment, IIASA, Laxenburg, 57-59.
Scholten, H. 2001. An ontological approach in quality assessment of modelling and simulation, in: A. W. Heemink, L. Dekker, H. d. S. Arons, I. Smit, and T. L. v. Stijn, editors. PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology,

Medicine and Biomedical Engineering. TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0, Delft, The Netherlands.
Scholten, H., B.J. De Hoop and P.M.J. Herman, 1990. SENECA 1.2: a Simulation Environment for ECological Application (Manual), DIHO, Yerseke, ISBN 90-74638-01-5, 150 pp.
Scholten, H. and A. Kassahun, 2006, Supporting multidisciplinary model-based water management projects: a user perspective, Summit on Environmental Modelling and Software, 3rd Biennial meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, USA, July 9-12, 2006.
Scholten, H., A. Kassahun and J.C. Refsgaard, 2006. Managing multidisciplinary model based water management projects. In: P. Gourbesville, J. Cunge, V. Guinot, S.-Y. Liong (Eds.), 7th International Conference on HydroInformatics, Nice, France, 4-8 September 2006. Research Publishing, Volume 3, 2231-2238.
Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.
Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.
Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.
Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.
Scholten, H., R. H. Van Waveren, S. Groot, F. Van Geer, H. Wösten, R. Koeze and J. J. Noort, 2000. Good Modelling Practice in water management. Proceedings HydroInformatics2000, International Association for Hydraulic Research, Cedar Rapids, Iowa, USA,
Scholten, H., R. H. Van Waveren, S. Groot, F. Van Geer, J. H. M. Wösten, R. D. Koeze, and J. J. Noort. 2001. Improving the quality of model-based decision support: Good Modelling Practice in water management. Pages 223-230 in A. Schumann, J. Xia, M. Marino, and D. Rosbjerg, editors. Regional Management of Water Resources. Proceedings of Symposium S2 of the 6th Scientific Assembly of the International Association of Hydrological Sciences, Maastricht, The Netherlands, July 18-27, 2001. IAHS, Maastricht, The Netherlands.
Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.
Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze, J. Noort,. Good Modelling Practice Handbook, STOWA, Utrecht, RWS-RIZA, Lelystad, The Netherlands (in Dutch, English version from http://informatics.wur.nl/research projects/pub-pdf/gmp.pdf), 1999.
Wittgenstein, L.J.J., 1921. Logisch-Philosophische Abhandlung. Annalen der Naturphilosophie, 14.
Zeigler, B.P., 1976. Theory of modelling and simulation. Robert E. Krieger Publishing Company, Malabar, Florida, ISBN 0-89874-808-9, 435 pp.

Appendix A Mathematical model details

Content of Appendix A, belonging to chapter 2

A.1 Mathematical modelling details

A.1.1 Format for mathematical models consisting of differential equations

A mathematical model consisting of differential equations can be described by the following state space representation of the state variables dynamics, leaving out spatial explicit processes, i.e. restricting to Beck's *Class II models*, in which the partial differential equations are approximated with a *finite element* or *finite difference*[1]*s* scheme (Beck, 1986):

$$\frac{dx(t)}{dt} = f\{x(t), u(t_k), v(t_k), \alpha, \beta\} + \xi(t) \qquad \text{Eq A-1}$$

$$x_0 = x(t=0) \qquad \text{Eq A-2}$$

$$y(t_k) = h\{x(t), u(t_k), v(t_k), \alpha, \beta\} + \eta(t_k) \qquad \text{Eq A-3}$$

With:

- x – vector of state variables
- x_0 – vector of initial values of the state variables x
- y – vector of output variables (i.e. model outcomes)
- u – vector of measured input variables
- v – vector of decision variables
- α – vector of fixed parameters
- β – vector of calibration parameters with probability density functions
- t – continuous time
- t_k – discrete instants in time at which outputs are observed
- ξ – vector of disturbances of the state variable dynamics that are not observable (system noise)
- η – vector of output observation errors (the measurement noise)

The *state space form* originates from control engineering (Van Straten, 1986, 1998, Beck, 1986) and can be defined[2] as a way to describe a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations[3].

A.1.2 OR models

A general form of Operation Research (OR) model is:

$$\frac{\max}{x \in X_0}(q = f(x) \in \Re^k) \qquad \text{Eq A-4}$$

$$X_0 = \{x \in \Re^n : g(x) \le 0 \in \Re^m\} \qquad \text{Eq A-5}$$

With:

- f – vector of objective functions $f_i(\mathbf{x})$
- g – vector of constraints $g_j(\mathbf{x})$

Operations Research is focused on various kinds of models, each with their own methods and solvers, including but not restricted to:

- *Linear Programming* studies the case in which the objective function f is linear and the set A is specified using only linear equalities and inequalities;
- *Integer Programming* studies linear programs in which some or all variables are constrained to take on integer values.

---

[1] *Finite element* and *finite difference schemes* are numerical methods to transform partial differential equations (PDEs) to ordinary differential equations (ODEs), which will not further discussed here. What is said on ODEs is equally appropriate for PDEs.
[2] From Wikipedia, http://en.wikipedia.org/wiki/State_space_(controls)#Controllability_and_observability.
[3] The terminology used here is defined in chapter 9.

- *Quadratic Programming* allows the objective function to have quadratic terms, while the set A must be specified with linear equalities and inequalities.
- *Nonlinear Programming* studies the general case in which the objective function or the constraints or both contain nonlinear parts.

A.1.3 An OR-approach for continuous simulation models

The approach chosen for continuous modelling and simulation models for environmental and ecological domains in this book, combines the OR-approach (optimization of one or more object functions within a series of constraints, as described in Eq A-4 and Eq A-5) with the mathematical approach described in Eq A-1, Eq A-2 and Eq A-3. Therefore the simulation model is used as a complex function and the objective function will be defined based on the distance between the model outcomes and the associated system observations. If this distance can be expressed in a single measure, i.e. the object function, calibration is an optimization problem, aiming at minimization of the object function by varying the uncertain model inputs, i.e. the uncertain parameters, the uncertain initial values of the state variables and decision variables.

There are many ways to represent the differences between model outcomes and observations for a single output variable in a so called *norm* or *criterion*, such as (for a single observed output variable):

Relative error: (Carver, 1980) $$c_1 = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{o_k}{y_k}-1\right)^2 \qquad \text{Eq A-6}$$

L_1-norm: (Tarantola, 1987): $$c_2 = \frac{1}{n}\sum_{k=1}^{n}|e_k|;\ with\ e_k = o_k - y_k;\ k = 1 \ldots n \qquad \text{Eq A-7}$$

L_∞-norm (Tarantola,1987): $$c_3 = \max\{e_1, e_2, \ldots, e_n\} \qquad \text{Eq A-8}$$

Sum of Squares for an individual output variable: $$c_4 = \sum_{k=1}^{n} e_k^2 \qquad \text{Eq A-9}$$

With:

c_i	–	i^{th} criterion (i.e. norm)
y_k	–	time series of observed output variables (i.e. model outcomes) at time instants at which outputs are observed
o_k	–	time series of observed output variable in the real system
e_k	–	k^{th}-residual, i.e. $e_k = o_k - y_k$
k	–	index for time instants at which outputs are observed
n	–	number of observations for a specific output variable

In the multivariate case (more output variables are observed) several criteria for a single output variable have to be combined in a single criterion. This can be done in many ways, e.g.:

The weighted sum of the individual sums of squares, e.g. Carver (1980): $$c_5 = \frac{\sum_{k=1}^{n} e_{1,k}^2}{w_1} + \frac{\sum_{k=1}^{n} e_{2,k}^2}{w_2} + \ldots + \frac{\sum_{k=1}^{n} e_{m,k}^2}{w_m} \qquad \text{Eq A-10}$$

Klepper (1989): $$c_6 = \max\left\{\frac{\sum_{k=1}^{n} e_{1,k}^2}{w_1} + \frac{\sum_{k=1}^{n} e_{2,k}^2}{w_2} + \ldots + \frac{\sum_{k=1}^{n} e_{m,k}^2}{w_m}\right\} \qquad \text{Eq A-11}$$

With:

c_i	–	i^{th} criterion (i.e. norm)
$e_{i,k}$	–	k^{th}-residual for i^{th} output variable , i.e. $e_{i,k} = o_{i,k} - y_{i,k}$
$y_{i,k}$	–	time series of i^{th} output variable at k^{th} time instant at which outputs are observed
$o_{i,k}$	–	time series of i^{th} observed output variable in the real system
$1/w_i$		weight for i^{th} observed output variable; if the a priori variance is known, its inverse can be used as weight
k	–	index for time instants at which outputs are observed
i		index for observed output variables
n	–	number of observations for a specific output variable

m number of observed output variables

Varying the decision variables corresponds here with drawing values for the uncertain factors, i.e. uncertain parameters (β), uncertain initial values of the state variables (x_0) and the decision variables (v). The constraints in the optimization model approach correspond here with drawing parameter vectors from their *a priori* probability density functions[4]. The same applies for the unknown initial values of the state variables (and their probability density functions). Decision variables in continuous modelling and simulation models for environmental and ecological domains are the model inputs for evaluating various scenarios and play a similar role as in optimization models.

A.2 References

Beck, M.B., 1987. Water quality modelling: a review of the analysis of uncertainty. Water Resources res. 23, 1393-1442.

Carver, M.B., 1980. Parameter optimization in the continuous simulation packages FORSIM and MACKSIM. Mathematics and Computers in Simulation 22, 298-318.

Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary, DGW-Ministery of Transport and Public Works, Wageningen University, Wageningen, 270 pp.

Tarantola, A., 1987. Inverse problem theory. Methods for data fitting and model parameter estimation. Elsevier, Amsterdam, 613 pp.

Van Straten, G., 1986. Identification, uncertainty assessment and prediction in lake eutrophication, Ph. D. Thesis. Technical University Twente, Febodruk Enschede, 240 pp.

Van Straten, G., 1998. Models for water quality management: the problem of structural change. Wat. Sci. Techn. 37, 103-111.

[4] The probability density functions reflect the scientific knowledge on the parameters or initial values by observation or scientific experiments.

Appendix B Structured modelling and simulation details

Content of Appendix B, belonging to chapter 2

B.1 Ad hoc modelling schemes

The scheme of Scholten and Udink ten Cate (1999) is one of many similar approaches to structure the modelling process (Figure B-1). It is one of a series published since 1994 in an attempt to make modelling more explicit. The first of these schemes (Scholten, 1994, Scholten and Udink ten Cate, 1995, 1996, 1999) belong to the category informal.

B.2 Good Modelling Practice

The seven steps of the first and the steps of the second decomposition level are shown in Figure B-2.

The Good Modelling Practice handbook has been written in a stepwise approach. It is based on the cross-fertilization between the multidisciplinary project team consisting of scientists, an active supervising committee with a more practical background, the strong commitment of the 50 participants in a workshop and, finally, the efforts of two series of experienced and inexperienced 'field testers' (users, not involved in developing the product).

The GMP handbook is a self-explaining document to support the entire procedure of the modelling and simulation process. It consists of a clear demarcation of the types and domains of models for which it is intended, a glossary of all concepts, an ontology of the modelling and simulation process, a checklist and summary, fill-in forms to document and archive the many steps and tests in the modelling and simulation process, the collective experience of large group of modelers on pitfalls and sensitivities in general and for specific modelling domains and finally references to specific literature and addenda on specific problems.

The backbone of the handbook is the informal ontology. At the highest level of the decomposition the steps are: (1) starting with a logbook, (2) defining the modelling project, (3) building the model, (4) analyzing the model, (5) using the model, (6) interpreting the results, and (7) reporting and archiving (Figure B-2). These steps are further decomposed to a level of a single modelling activity (e.g. determining for which factors the model is most sensitive or checking if all model objectives are met). The core part of the handbook includes also a series of tests without prescribing with which methods or with which algorithms these tests have to be carried out. These tests comprise of conceptual model validation, some aspects of verification (dimension check, mass or energy balance control), a robustness test, a sensitivity analysis, a calibration, a (historical data) validation, and an uncertainty analysis. In this way the handbook supports all activities related to modelling and simulation and passing these tests improves the credibility of the model.

In modelling and simulation for water management quite a number of different stakeholders are involved. These include the customer or client, who has to pay for the job and decides on the requirements and acceptance, the model builder, domain experts, the project manager, the model user (often called modeler), the policy/decision maker, the (internal) tester, and the (external) auditor. All these actors can profit of the handbook, whether in formulating simulation projects, recording its progress, assessing its quality, or otherwise.

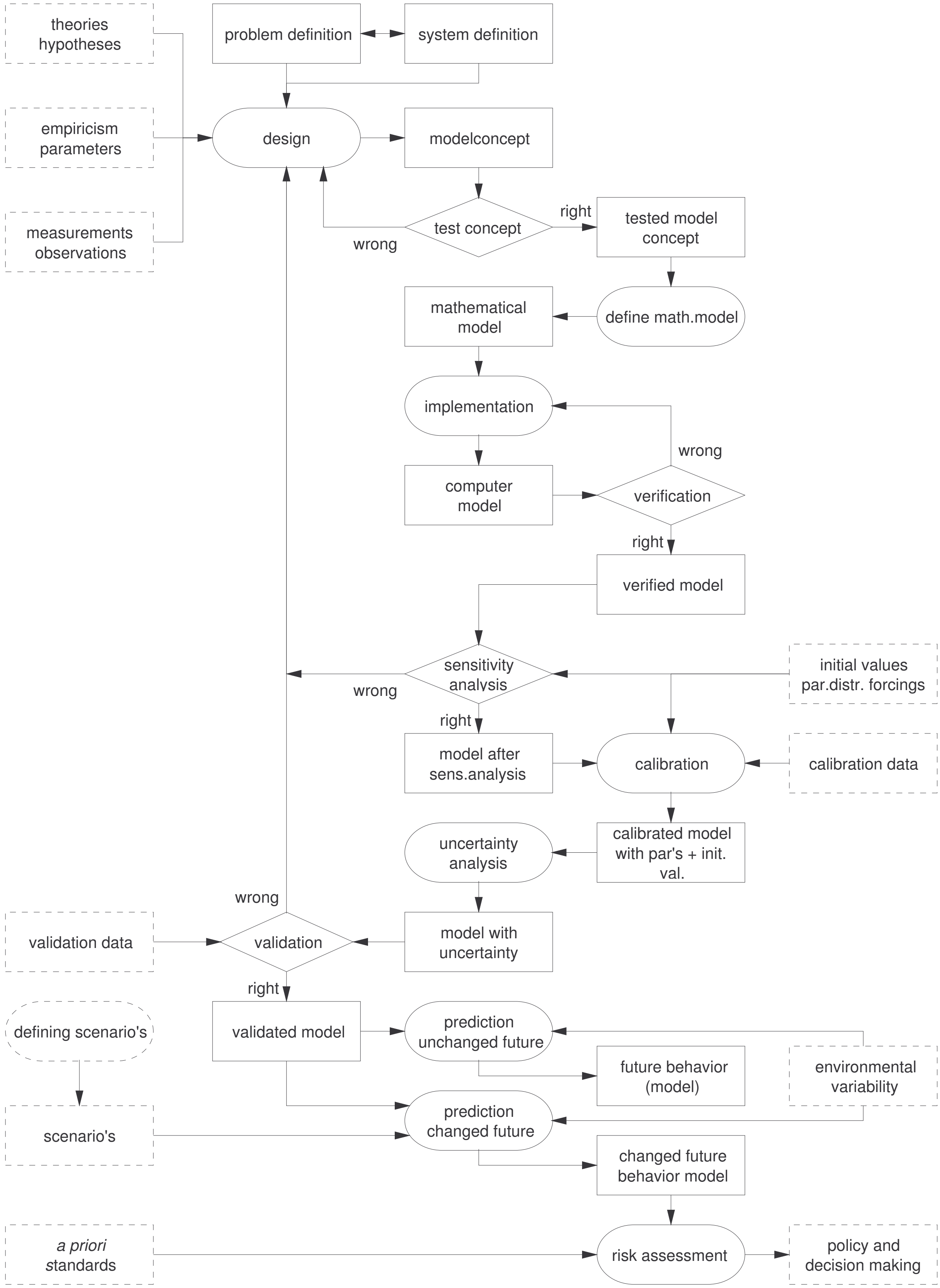


Figure B-1. Instance of a structure for the modelling and simulation process. Rectangles are products, ovals actions, diamonds actions with a decision aspect. Dotted lines are left out (from Scholten and Udink ten Cate, 1999).

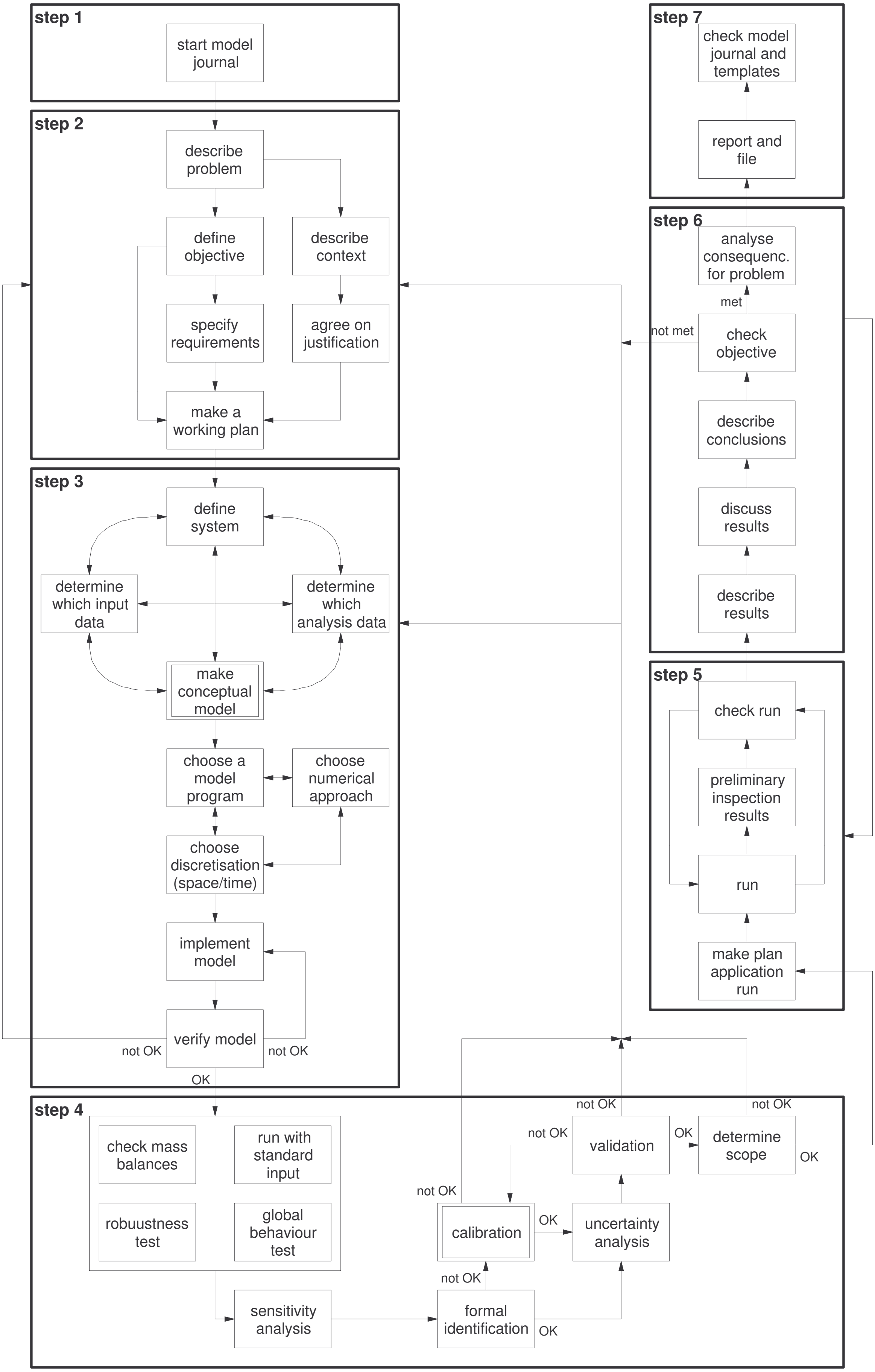


Figure B-2. The first two decomposition levels of the modelling process in the GMP handbook (Van Waveren *et al.*, 1999).

B.3 HarmoniQuA

A book form, i.e. of the GMP Handbook and the Dutch norms for modelling for water management, is not an appropriate format for guidelines. In addition, the Dutch GMP handbook and norms have several other disadvantages. They are typically national (Dutch) initiatives and products with a strong link to what is internationally referred to as the 'polder model', a typically Dutch way of decision making in management, which is fundamental for norms in general. This is a good development, sharing knowledge, accepted by a large group, forms also the base for an ontological approach proposed in this book, but the national character is also a shortcoming, as its use is restricted to modelers for water management in the Netherlands.

Furthermore, EC directives, e.g. the Water Framework Directive (WFD), enforce countries to co-operate at catchment level and to take water management decisions based on integrated studies, covering all relevant subdisciplines of model based water management, especially including the socio-economic aspects. The EC also recognize the key role of models to solve water-related problems nowadays. Co-operation in this sense requires that several boundary conditions are fulfilled. First there are the legal and organizational issues of implementing WFD in each of the EC countries. Moreover, models used on two sides of a border between countries should seamlessly be applied together, which sets demands to the software of these models. The latter kind of model coupling problems is largely solved in the EC funded project Harmon-IT (Blind *et al.*, 2005, Gijsbers and Gregersen, 2005). Finally, a harmonization of modelling methodologies for water based water management in the different EC countries is needed as well as a match between modelling methodologies in use in different subdisciplines of model based water management.

The objectives of the HarmoniQuA project are:

1. developing a common, generic methodology, terminology and guidelines with aspects, specific for the domains of water management;
2. supporting multidisciplinary model-based water management by developing a knowledge base, building a Modelling Support Tool (MoST);
3. testing these products;
4. extending the project results to the wider professional community;
5. raising awareness to use the developed methodology and tool to facilitate quality assurance and make modelling projects more transparent.

The scientific achievements were:

1. Defining the modelling process, make modelling expertise explicit and structure it in an ontological Knowledge Base (KB) with modelling knowledge and a glossary with terminology; the explicit nature and the fact that it is shared by a substantial part of the professional modelling community represents the main scientific achievement.
2. Using state-of-the-art knowledge engineering techniques (ontologies implemented in Protégé) to structure the modelling knowledge.
3. Developing a Modelling Support Tool, (MoST) to support teams in multidisciplinary model-based water management by:
   - Providing the explicit modelling knowledge, filtered for relevant water management domains and the role of the user in the team;
   - Setting up modelling projects for co-operation in (distributed) teams;
   - Monitoring and recording what all team members do in a project;
   - Helping to write reports on performed activities and decisions for various purposes and different audiences.
4. Helping users to:
   - Edit the KB for maintenance and upgrading with a KB Editor. The KB Editor act as front-end for Protégé and is easy to use for domain experts, untrained in knowledge engineering techniques;
   - Get familiar with the complex methodology and the Modelling Support Tool (MoST) by providing extensive multimedia training material and help system.

The main deliverables are an integrated product consisting of the following elements:

1. A Knowledge Base with guidance for multidisciplinary model-based water management and more than 700 glossary entries, both implemented in Protégé;

2. Modelling Support Tool, MoST;
3. KB Editor, i.e. a Protégé front-end for domain experts unskilled in knowledge engineering;
4. Websites (public, training, help);
5. Multimedia training material.

These HarmoniQuA objectives, scientific achievements and main deliverables are discussed in more detail in chapter 6 (objective 1, achievements 1 and 2 and deliverable 1) and chapter 7 (objective 2, achievements 3 and 4 and deliverable 2). Objective 3 is discussed in chapter 10, while objectives 4 and 5 and deliverables 3, 4 and 5 are hardly or not discussed in this book[1].

B.4 References

Blind, M.W., R.V. Moore, H. Scholten, J.C. Refsgaard, S.A. Borgvang, C. Giupponi, I. Borowski, M. Estrela, D.G. George, J. Froebich, I. Zsuffa, P. Vanrolleghem and W. De Lange, 2005, Current Results Of The EC-sponsored Catchment Modelling (CatchMod) Cluster. In: A. Zerger, R.M. Argent (Eds.), MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, December 2005, Modelling and Simulation Society of Australia and New Zealand, ISBN: 0-9758400-2-9, 1210-1216, http://www.mssanz.org.au/modsim05/papers/blind.pdf.

Gijsbers, P.J.A. and J.B. Gregersen, 2005, OpenMI: A glue for model integration. In: A. Zerger, R.M. Argent (Eds.), MODSIM 2005 International Congress on Modelling and Simulation, Melbourne, December 2005, Modelling and Simulation Society of Australia and New Zealand, 648-654, http://www.mssanz.org.au/modsim05/papers/gijsbers.pdf

Scholten, H., 1994, Quality assurance of the simulation modelling process, A.R. Kaylan, A. Lehmann, T.I. Ören, Proc. of the European Simulation Symposium, Istanbul, October 9-12, 1994, Istanbul, Vol. I, 85-89.

Scholten, H. and A.J. Udink ten Cate, 1995. Testing simulation modelling quality, Mathematical modelling and simulation in agriculture and bio-industries. IMACS/IFAC, Brussels, Belgium, pp. II.A.3.1-6.

Scholten, H. and A.J. Udink ten Cate, 1996. Quality assessment of simulation modelling. In: C. Lokhorst, A.J. Udink ten Cate, A.A. Dijkhuizen (Eds.), Information and Communication Technology applications in Agriculture: state of the art and future perspectives. ICCTA, Wageningen, pp. 353-358.

Scholten, H. and A.J. Udink ten Cate, 1999. Quality assessment of the simulation modelling process. Computers and Electronics in Agriculture 22, 199-208.

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J.J. Noort, 1999. Good Modelling Practice Handbook. STOWA/RWS-RIZA, Utrecht/Lelystad, original in Dutch (Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice), ISBN 90-5773-056-1, 149 pp.

[1] See for instance the publication list: http://harmoniqua.org/public/Product/papers.htm.

Appendix C Requirements to a process KB and a KB-editor

Content of Appendix C, belonging to chapter 6

C.1 Introduction

In chapter 6 the requirements to the ontological knowledge base (KB) for process and instantiated for modelling have been discussed, as well as design criteria for the Knowledge Base Editor (KB-editor). Here more details are given, especially on the principles leading to design criteria (given in Table C-1), the requirement analysis method and criteria classification (section C.2) and on how the criteria are used in the implementation of the KB and KB-editor.

Table C-1. Decomposition principles for the process ontology, used as starting point for the design characteristics in section C.3.

Decomposition principles process ontology

1. A decomposition should be practical, i.e. a *piece of work* should not be decomposed into too many *pieces of work*;
2. A decomposition should not have too many decomposition levels, in order to be of practical use;
3. Each *piece of work* (a 'concept' in ontological terms) should match with the perception of professional modelers and fit in their *performance repertoire* of professionals;
4. Each *piece of work* should be well defined and concise enough to be perceived by professionals as single *pieces of work*, functionally connected with each other representing the whole modelling process;
5. A single *piece of work* should have one or more clear inputs and one or more clear outputs;
6. The level of detailing should be sufficient in order to be concrete and small enough to be practical and acceptable for its intended users.

C.2 Requirement analysis method

A typical classification for software requirements is described in Kotonya and Sommerville (1998). Their classification distinguishes *functional* (what the system should do) and *non-functional* requirements (constraints). The non-functional requirements concern (1) the process of software engineering, (2) the product to develop and (3) external requirements. Although often used, this classification of requirements is sometimes ambiguous. Glinz (2005) proposed a classification, in which four facets to requirements are distinguished:

1. *Kind*: including functional requirements, but also performance and others;
2. *Representation*: forms in which requirements are represented;
3. *Satisfaction*: requirements can be hard (i.e. yes or no fulfilled) or soft (i.e. gradual);
4. *Role*: requirements play a role by (1) specifying properties of the system, (2) describing facts or rules of the system environment that influence the system and (3) specifying the behavior of actors in the system.

The **kind facet** has five possible values: *function* (what the system should perform), *data* (data item or structure that should be part of the system), *performance* (related to events, reaction time, intervals, speed, volumes, rates, etc.), *specific quality* (product qualities (e.g. reliability, usability) and product management qualities (e.g. maintainability, portability)) and *constraint* (external constraints and design decisions of stakeholders etc.).

The **representation facet** has four possible values: actions to perform and data to be provided are represented in an *operational* form and can be verified formally or by testing, performance requirements have to be represented in a *quantitative* form, requirements at a more abstract level (e.g. business goals and usability goals) are represented in a *qualitative* form and requirements describing a required situation are represented in a *declarative* form.

The **satisfaction facet** has two possible values: the *hard* (yes/no) type of requirements and *soft* requirements that can gradually be fulfilled. The more effort is spent, the more the requirement will be satisfied.

The **role facet** has three possible values: the 'classic' requirements to the system are *prescriptive*. *Normative* requirements refer to norms to the system set by the system environment. Requirements to the behavior of actors using the system are called *assumptive* and they are out of control of the system.

An overview of this requirement classification of Glinz (2005) is shown in Figure C-1. It will be used to classify requirements for the process knowledge base (see Table C-2).

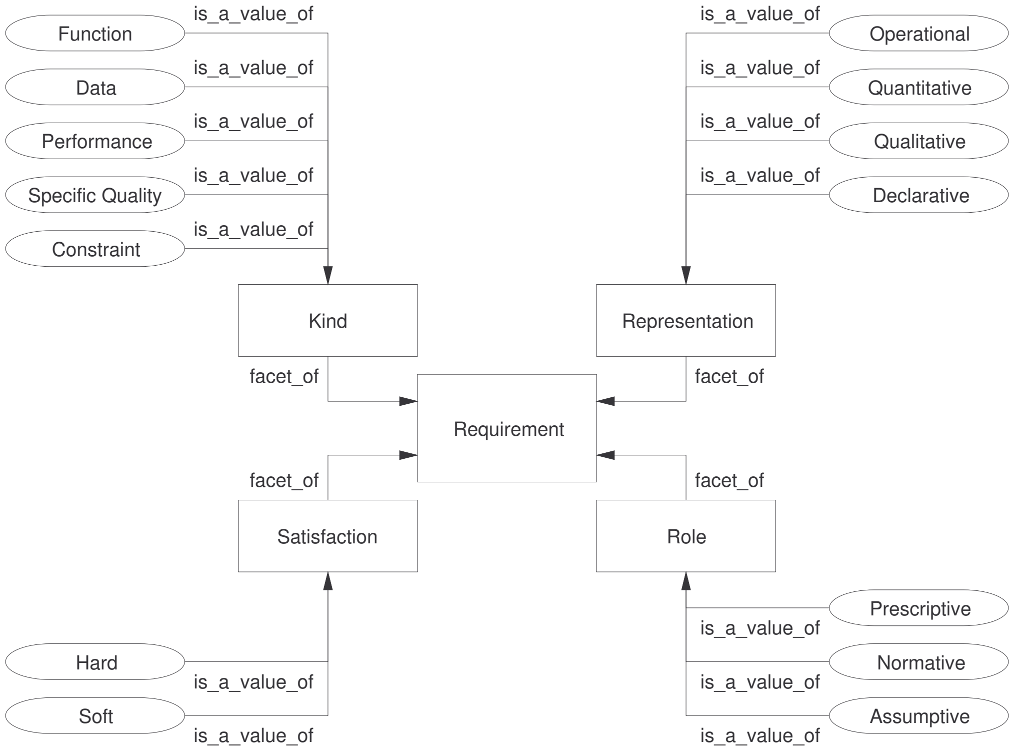


Figure C-1. Requirement classification according to Glinz (2005).

C.3 Design requirements to a process knowledge base and associated tools

The list of requirements presented in Table C-2 is an elaboration of the decomposition principles and a series of wishes from users of preliminary (β) versions of the KB, the KB-editor and the Modelling Support Tool.

Table C-2. Requirements to the KB and KB-editor, categorized according to Glinz (2005). The *requirements* in the left column are from chapter 6. The *design solutions* (middle column) explain the requirement and indicate how the requirement has been realized. The *categories* (right column) classify the requirement according to Glinz (2005).

Requirement	Design solution	Categories
1. The KB should make knowledge explicit	This is one of the side-objectives of this book. Choosing for an ontological approach makes the knowledge explicit, which should not be confused with 'formal'.	• Data • Declarative • Hard • Prescriptive
2. A proper granularity (i.e. level of detailing) of the knowledge in the KB should be chosen	The level of detailing will be chosen carefully, trying not to give too many details to avoid being too prescriptive and not too less to avoid being too abstract and vague. The proper level of granularity should be tested and evaluated, as it is difficult to make this requirement measurable. See also chapter 6, section 6.4.3.	• Data • Qualitative • Soft • Prescriptive
3. KB should be flexible (i.e. easy to change its structure)	When the ontological structure is changed the ontological development tool has to protect the integrity and consistency of the KB. A proper ontology development tool enables the KB to be flexible.	• Function • Operational • Hard • Prescriptive
4. Maintenance of the KB should be easy	Using he ontology development tool Protégé for maintenance of the ontology (the KB structure) and its instances (i.e. the KB) will enable easy maintenance of the ontology.	• Specific quality • Qualitative • Soft • Prescriptive

5. Updating of the knowledge in the KB should be easy	The ontology tool should enable easy and safe updating the knowledge in the KB. Protégé facilitates updating of by knowledge engineering experts. Although Protégé is easy to use and especially designed for disciplinary knowledge experts, inexperienced in knowledge engineering, it was necessary to develop a web based front-end Knowledge Base Editor.	• Specific quality • Qualitative • Soft • Prescriptive
6. Adding and editing the KB should be protected by an adequate authorization system	The web based front-end Knowledge Base Editor should contain a full knowledge updating authorization system, allowing unauthorized persons to give comments on (pieces of) the knowledge base, persons responsible for a discipline to edit, add and delete the disciplinary parts of the knowledge and only knowledge administrators full access. This safety system is completed with automatic saving versions of the KB, keeping these available and making back ups. In this way any change in de the KB can be undone (see also requirement 7for the organizational design aspects).	• Function • Operational • Hard • Prescriptive
7. Some authority or board should control the content of the KB.	In order to ensure that the content of the KB is accepted by large parts of the professional community some institution should be responsible for its content. This can be a user panel, a Community of Practice[1]. Or a similar organization.	• Constraint • Qualitative • Soft • Assumptive
8. The KB should be 'open', allowing all interested persons to comment on the KB	The authorization system (see 6) should allow anyone to read the content KB and comment on it in order to get acceptance by a larger group of experts. Persons, unauthorized to edit the KB, can give comments by the web based Knowledge Base Editor. Furthermore, the openness of the KB should assure that scientific progress (new methods, tools, methodology) is not blocked. In an optimal design of this feature, users should be allowed to add new knowledge, first at a personal or project level, later as part of the modelling knowledge base available for all, after approval by the KB authority board, see 7.	• Function • Operational • Hard • Prescriptive
9. The part of the KB that has to be 'understood' by computers (the ontological structure) should be as formal as possible.	Any ontology should be made as formal as possible, but if computers/machines have to handle and 'understand' the knowledge, this is essential and will be realized.	• Data • Qualitative • Soft • Assumptive
10. The part of the knowledge base that has to be understandable for persons should be less formal and more textual.	The requirement of the level of formalism (see 9) may be relaxed if not computers/machines have to understand it; a lower level of formality is simpler to achieve than a level in which the knowledge is completely formal. Being understandable for humans is sufficient.	• Data • Qualitative • Soft • Assumptive

[1] A CoP (Community of Practice) can be defined as a group of (distributed, e.g. over different departments) professional practitioners, who in, possibly informal, relation with each other have a shared domain of interest or have similar task responsibilities, that, over time, develops a shared meaning and increases the spreading of knowledge and information among its members (Markestijn, 2004).

11. The more basic ontological layers (layer 0, layer 1 and perhaps 2; see Figure 4-6 and requirement 21) of the ontological structure should be reusable for other processes.	The primary subject of the KB is the modelling process, but the results so far (and discussed in the remainder of this chapter) indicate that parts of the KB (especially its structure) are useful for completely other processes, e.g. business processes, supply chain management, medical procedures, procedures for professional healthcare. Therefore two process defining ontological layers are used: layer 1 the *generic process ontology* and layer 2 the *specialized process ontology instances*. The process of modelling resembles in most aspects any other process, but its content (the things to do) are rather complex and require much expertise; like many workflow management systems there are several phases to distinguish: the initial phase defines (or adapt) the structure of the process, the next phase is focused on performing what has been defined; processes that have to be proceed in a similar way (defining it and performing what has been defined) can be fit in the KB structure designed.	• Data • Qualitative • Soft • Assumptive
12. The KB should be consistent.	Although this requirement is not critical[2] for the part of the KB, which does not have to be understandable for computers/machines, it is very important but hard to fulfill; some of the consistency aspects are solved by Protégé, the tool to build ontologies, while the remaining inconsistencies have to be identified and solved by experts in modelling (careful editing of the knowledge and by testing the KB content in practice; see also chapter 10).	• Data • Qualitative • Soft • Normative
13. The KB should be complete.	The KB should contain knowledge on all aspects of the 'things to do'[3]; on the other hand it will not be complete in the sense that all methods to use will be included; it should be allowed that new approaches and methods from research can be added or used, even if they are not included by the KB. Completeness of the KB consists of two aspects: (1) identifying which 'things to do' belong to the process at hand, e.g. modelling and (2) which level of detail is required and should the knowledge be complete at the lowest level of decomposition; the latter means that for all activities there are methods to help in doing the activities; there always new methods or methods that are obsolete, because newer and better (i.e. more effective and/or more efficient are available).	• Data • Qualitative • Soft • Prescriptive

[2] 'Critical' means here: human users can use the KB, despite inconsistencies, but between computers/machines inconsistencies will lead to erroneous results.
[3] The term 'things to do' is intentionally used to avoid terms as 'task' and 'activity', which are protected terms that will be (later) used with a more precise meaning.

14. The knowledge should be shared by a substantial fraction of the professional community.	An ontology should (by definition) be used to share knowledge among a group of experts, e.g. modelling experts; an ontological approach has therefore always the aim of sharing the knowledge. Here, sharing of the process knowledge by substantial groups of the professional community will be realized by a process of instantiating an ontological modelling KB, of which an actual example will be discussed in chapter 6, section 6.6.4 and chapter 7.	• Data • Declarative • Soft • Assumptive
15. The KB and the support tool should reflect the work of (distributed) teams working in multidisciplinary model-based projects with different purpose types and of different complexity.	Throughout this book (and especially in chapter 4) it has been said that modelling is nowadays often carried out in multidisciplinary projects, in which teams have to use models to solve multidisciplinary problems. The multi-user (i.e. working in a team) and multidisciplinary requirements will be realized by: • The architecture, being client-server with all shared 'records' and the guidelines on the server; • Organizing the knowledge in the knowledge base on what to do by customizing this according to the following aspects: o Roles or user types (see requirement 16); o Purposes (not implemented; see requirement 17); o Domains/disciplines (see requirement 18); • Project complexity (see requirement 19);	• Specific quality • Qualitative • Soft • Normative
16. The KB should enable different roles in the team.	Team members work in different roles in the team with different responsibilities and different things to do; therefore they need different knowledge in the form of guidance and different support in their work. The latter is related to the Modelling Support Tool and will be discussed in chapter 7. In many modelling project the following roles can be distinguished: • *MANAGER* (client), • *MODELER*, • *AUDITOR*, • *STAKEHOLDER*, • *PUBLIC*. In other processes other roles are relevant; roles have therefore to be defined in the modelling process instance (i.e. in layer 2 of Figure 4-6; see also chapter 6, section 6.6.3).	• Data • Declarative • Soft • Prescriptive
17. The KB should enable different purpose types.	Problems have a context in which they should be solved. There many different purpose types of running processes in a project, of which *PLANNING*, *DESIGN*, *OPERATIONAL MANAGEMENT* are relevant for modelling.	• Constraint • Declarative • Soft • Normative

18. The KB should enable team members, from different domains and disciplines, to cooperate in synchronous and asynchronous subprojects.	By definition multidisciplinary projects use knowledge originating from different disciplines/domains; team members responsible for a discipline/domain need different knowledge than those working in other domains; sometimes the work in different disciplines can be or has to be executed simultaneously or with the same speed (synchronously), while the work for some other discipline depends on the results of the work in a third discipline and has therefore to be performed in a serial order (asynchronously). In chapter 6, section 6.6.4 this is instantiated for multidisciplinary model-based water management with the following (water management) domains: *water quality*, *precipitation-runoff*, *ecology*, *groundwater*, *river hydrodynamics*, *flood forecasting* and *socio-economics*.	• Data • Operational • Hard • Prescriptive
19. The KB should enable different degrees of project complexity.	A simple or routine multidisciplinary project, realizing a collaborative process, e.g. applying a model for a routine water management problem, has a less complex structure, in which parts of the work can be skipped or combined with other things to do. There are big differences in the complexity of such projects. They range from routine projects with only a few tasks to do and more academic or research oriented projects in which all tasks have to be done. Therefore three levels of predefined project complexity are distinguished, *BASIC*, *INTERMEDIATE* and *COMPREHENSIVE*. In the project-defining phase of a project (during project initialization or later) these templates of tasks to be done can further be adapted to the demands of the project at hand.	• Data • Operational • Soft • Prescriptive

20. The KB should be suited for processes in general including, but not restricted to various types of simulation and applying mathematical models[4].	The design of the instance of the KB (chapter 6, section 6.6.4) is focused on simulation models for water management, but the KB aims at a wider applicability, being useful and relevant for other types of mathematical models and even other process. Therefore it will be designed in 'ontological layers' (see Figure 4-6) with increasing specialization. This will allow re-use of the more general parts for other types of processes in which teams have to work in multidisciplinary projects. Furthermore, the terminology used in each ontological layer should correspond with its level of specialization. In this way, specialized pieces of knowledge will be separated from more generic) parts by allocating (sub)domain/(sub)disciplinary parts according to the customization property 'domains/disciplines' (see below). It should be tested to what extend the KB is suited for other mathematical models and for other processes (see chapter 10).	• Specific quality • Qualitative • Soft • Normative
21. The KB should be layered in a way that the most generic process knowledge is in a more generic ontological layer[5] (with a lower number) and the most specific knowledge in the more specialized ontological layers (with a higher number), i.e. the instances of the ontology.	This practicality enables reuse of the higher ontological layers of the KB for other process than modelling. This has been introduced in chapter 4 (see Figure 4-6) and will be discussed further in chapter 6, section 6.6. Overview of ontological layers: • Layer 0: process, project and other terminology • Layer 1: generic process terminology • Layer 2: specialized process instances, e.g. modelling, business process, planning • Layer 3: more specialized process instance, e.g. modelling for water management, supply chain management, water stress mitigation • Layer 4: project journal structure of a process, described in ontological layers 2 and 3, e.g. for modelling • Layer 5: instances of layer 4, i.e. filled project journals of actual projects.	• Specific quality • Declarative • Hard • Prescriptive
22. The developed KB and the KB Editor should be platform (operating system) independent.	MS Windows™ is the prevailing operating system platform, but many professionals use other platforms, most Unix (or its variants). The Knowledge Base Editor has been developed in Java as web application and Protégé is free, open source software developed in Java. This guarantees platform independency.	• Specific quality • Declarative • Hard • Prescriptive

[4] 'Mathematical models' are defined in chapter 1 as: *models that share the use of mathematics to represent relevant parts of an object system to solve some problem(s) related to the object system. The models range from discrete or continuous simulation models to operation research models.*
[5] The concept 'ontological layer' is discussed in the meta-ontology of chapter 5; see also Figure 4-6.

23. The Knowledge Base Editor has to have to following functionality: • Set-up a new KB (supporting new types of processes); • Edit a KB which is developed based on the ontological structure described in this chapter; editing includes adding new aspects, changing, deleting, reading, commenting, etc.; • It should input the knowledge and its changes in the Protégé KB in the proper format; • There should an authorization system for a KB with the following types of users and privileges: o General users: may review and comment content of the KB; o KB editors: may edit the content of the KB; o KB administrators: may authorize users to edit and manage users and their privileges. • Version management of the KB; • Editing glossaries; • Registering users; • Enabling downloading KB and Modelling Support Tool.	Knowledge experts find it difficult to use Protégé to enter their knowledge in the KB. The Knowledge Base Editor should work as a front-end for Protégé for this type of users (see also requirements 4, 5, 6 and 8). This tool has to be developed as web application, which implies that it is available for all operating systems, always and everywhere, for users connected to Internet.	• Function • Operational • Hard • Prescriptive
24. The KB should be designed in way that its ontological format does not hinder moving to other formats.	Ontologies are stored in a format; some ontological formats have more intrinsic representation power than others; switching from one format to another may have undesirable consequences, i.e. loosing information. This aspect is discussed in chapter 3.	• Specific quality • Declarative • Soft • Prescriptive
25. The tool to develop the ontology should provide all basic functionality that can be expected from such a tool, including handling of names of concepts and relations, allocating identification numbers to all concepts and relations, basic terminology.	Protégé has been chosen and this tool provides in the required functionality, see chapter 3, where this aspect has been discussed in more detail.	• Function • Operational • Soft • Prescriptive
26. Inexperienced knowledge experts should be able to upload their expertise to the KB.	The Knowledge Base Editor is an extension of the functionality of Protégé to enable domain experts to enter their expertise to the KB.	• Function • Operational • Soft • Assumptive

27. The server should contain all shared information (i.e. KB with guidance and the work done by the team).	The KB will be updated from time to time and therefore it should be kept available at a central server with local copies that can easily be updated by downloading a newer version from the server. In a client-server architecture as described before, this can easily be realized. Results of multidisciplinary projects, i.e. the work done, should be safely stored and available for those team members that are authorized to use the work of others. This will be realized by defining an ontological structure for process journals. This ontological structure will resemble the ontological structure of the KB, but with some extensions. Process journals will be stored on some server (LAN, web server, desktop computer or notebook) to facilitate that changes made by different team members are integrated to a single modelling journal, available for all authorized team members.	• Data • Operational • Hard • Prescriptive
28. Adding and editing knowledge (i.e. guidance stored in the KB) should be carried out on the client side and stored on the server.	This is typically the work of experts and therefore it should be performed at the client side with easy access controlled by an authorization system. With the web based Knowledge Base Editor this will be realized.	• Function • Operational • Hard • Prescriptive

C.4 References

Glinz, M., 2005, Rethinking the Notion of Non-Functional Requirements, Proceedings of the Third World Congress for Software Quality (3WCSQ 2005), Munich (D), 55-64, http://www.ifi.unizh.ch/groups/req/ftp/papers/3WCSQ2005.pdf.

Kotonya, G. and I. Sommerville, 1998. Requirements Engineering: Processes and Techniques. Wiley, New York, ISBN: 0-471-97208-8, 294 pp.

Markestijn, J., 2004. Communities of practice and supporting software, PhD, Artificial Intelligence, Knowledge Management and Knowledge Technology, Faculty of Sciences, Vrije Universiteit, Amsterdam, 97 (+XV) pp.

# Appendix D	Concepts of the process ontology

Content of Appendix D, belonging to chapter 6

D.1 Ontological layer 1: generic process knowledge

In Chapter 6 a process ontology for multidisciplinary model-based problem-solving has been proposed, which is structured in so called *ontological layers* aiming to facilitate reuse of parts of the ontology. *Ontological layer 1* contains generic process knowledge. In this section all concepts used in that ontology are defined in Table D-1. It contains all concepts depicted in the structure diagram of Figure 6-7 of Chapter 6.

Table D-1. Concepts of the process ontology, ontological layer 1, with generic process knowledge. All terms in italic are defined components of the meta-ontology (Chapter 5) or of the generic process ontology. Concepts are left aligned and concept properties right aligned.

Concepts	Meaning
Guideline	A detailed plan or explanation to guide you in setting standards or determining a course of action. Note 1: From WordNet 1.7.1. Note 2: Wikipedia: *any document that aims to streamline particular processes according to a set routine. By definition, following a guideline is never mandatory.*
Sub*project*	Part of a *project*, which runs relatively independent from other *subprojects*.
Step	Group of associated tasks. Note 1: The association between the *tasks* is reflected in the structure diagram in Chapter 6, Figure 6-6.
Generic *Task*	A complete part of a *process* (or of its instantiated *project* / *subproject*), which completion helps to accomplish the *process*. It is the smallest purposeful part of a *process*, but it is often a *composite action*, as it consists of one or more *activities*. Note 1: *Tasks* are related to achieving the *process*; have to do. Note 2: 'Purposeful' means here related to the purpose of the *process* / *(sub)project*.
(Ordinary) Task	*Task*, which is not a *decision task* or a *review task*.
Decision *Task*	An ordinary *task*, but it includes taking a decision on continuation; in case of a positive decision the process will continue with the *next task* and in case of a negative decision to a *previous task*.
Review *Task*	A *decision task* with emphasis on the interaction between different *roles* in the *multidisciplinary team*.
Method	From Greek μεθοδοσ, literally 'way across'. In science in general: a codified series of *actions*, taken to complete a certain *task* or to reach a certain objective. Note 1: From www.Wikipadia.com. Note 2: The term 'tasks' is here used in the common sense meaning; it includes both *tasks* and *activities*. Note 3. A *method* is a task element.
Applicability of *Method*	Indication for which of the instantiated *tasks* and *activities* the method can be used.
Task *element*	Characteristics describing relevant aspects of a task, including the *concepts 'activity'*, '*method*' .
Task Name	Textual identifier of a *task*. Note 1: Next to the name, Protégé adds automatically a numeric identifier to all concepts and relations, including *steps*, *tasks* and *activities*.
Task Definition	Formal description of a *task*.
Task Explanation	Informal description of a *task* clarifying what it consists of.
Activity	*Task* element describing what a team member has to do to accomplish (a part of) a *task*. An *activity* is a *single action* and the smallest part of a *process*. Note 1: *Activities* are related to what *team members* have to do.

Activity name	Textual identifier of an *activity*. Note 1: Next to the name, Protégé adds automatically a numeric identifier to all concepts and relations, including *steps*, *tasks* and *activities*.
Activity List	List of *activities* belonging to a *task*.
Activity Introduction	Introductory text for the *activities* belonging to a *task*.
Activity Description	Description of what has to be done in an *activity*.
Sensitivity and Pitfall	*Task element* indicating difficulties to be expected in a *task*. Note 1: In many cases *sensitivities and pitfalls* are associated with software used to do the *task* at hand.
Reference	*Task element* pointing to an external scientific paper / report / etcetera or to an Internet website. *Relation* denoting a pointer (including hyperlink) to a paper, book or report with a *scientific* discussion on (aspects) of a *task* or *activity*.
Software Aspect	*Task element* that refers to some (external) software used to do (a part of) the *task*.
Input	Product(s) that are needed to start the *task*.
Output	Product(s) that are expected at *task* completion.
Customization characteristic	Characteristic to make a task or activity more specialized. Note 1: These will be instantiated in *ontological layer 2*, see .Table D-2.

Table D-2 contains the extra terms of the generic process ontology from Figure 6-8. Table D-2 instantiates the concept *customization characteristic* of Table D-1 by defining the extra terms. The *customization characteristic domain* represents the multidisciplinary aspects to the knowledge in the KB. Two *instances* of *domain* (*generic domain* and *multi-domain*) are (rather) generic *concepts* of modelling and belong therefore to *layer* 1 (specialized process ontology modelling). The other *instances* of the *concept domain* are model-based water management related and belong therefore to *ontological layer 3* (defined in D.4). The *instance modeler* of *concept role* is of *ontological layer 2* (specialized process ontology for modelling), but is still presented in this section, dealing with *ontological layer 1*.

Table D-2. The (extra) concepts of ontological layer 1 (generic process ontology), extending the concept customization characteristic, using the meta-ontology of Chapter 5. All terms in italic are defined components of the meta-ontology (Chapter 5), or of the generic process ontology (section D.1). All concepts are also (sometimes slightly different) defined in the HarmoniQuA Glossary, which is a part of the HarmoniQuA KB for model-based water management (available on CD-ROM or through the public HarmoniQuA website: http://harmoniqua.wau.nl/public/Product/software.htm.

Concepts	Meaning
Role	This *customization characteristic* refers to the *role* of a member of a *multidisciplinary team* in a *process*-oriented *project* to solve *problems* using *models*.
Manager/client	*Role* of the person in a *project* that is the *problem owner* contracting a *multidisciplinary team*.
Expert	Role of the person in a *project* that has to provide (scientific) expertise. Note 1: The *concept expert* is a *role* and belongs to *ontological layer 1* (generic process ontology). It is more generic than *modeler*, which is an instance of *expert*.
Modeler	Role of the person in a *model*-based *problem-solving project* that has to do the *modelling*. Note 1: The *concept modeler* is an *instantiation* of the concept *expert*; *expert* belongs to *ontological layer 1* (generic process ontology) and *modeler* belongs to *ontological layer 2* (specialized process ontology for modelling).
Auditor	*Role* of the person in a *project* that has to do audits.
Stakeholder	*Role* of the person in a *project* that will be affected directly.
Public	*Role* of the person in a *project* that is generally interested.

Domain	(Scientific) field of application. Furthermore distinctions have been made between *domain*, *discipline* and *paradigm*. Note 1. Domain is defined in the meta-ontology of Chapter 5). Note 2. Domain will not be used in the sense of 'domain of (intended) application' (Schlesinger, 1979); instead the concept *model scope* will be used. *Model scope* is defined in Chapter 9 and Appendix G. Note 3. *Domain* and *subdomain* are used here more or less as synonyms. In fact 'water management' has to be seen as *domain* and *specializations* of water management (e.g. *hydrodynamics*, *groundwater*) as *subdomains*, but in HarmoniQuA these *subdomains* are called *domains*.
Generic *domain*	The *generic domain* covers the general methods that are not specific to single *domains*. The *generic domain* is the backbone of the Knowledge Base. The Knowledge Base of the single *domains* is for some *tasks*/*activities*/etc. identical to the *generic domain* while it other places contain variations of importance for the specific *domains*.
Multi-domain	The term multi-*domain* is used when a single *subproject* covers more than one *domain*. In that case, *model*ling these *domains* will typically occur synchronously. If *modelling* in more than one domain cannot be done synchronously, the work should be split in more than one *subproject*, which can run at different speeds. Note 1: *Tasks* specific for *multi-domain* contain work triggered by combining more than one *domain* (i.e. discipline). Note 2: Using *multi-domain* as a single *domain* has no sense (only to see which tasks and activities are involved); *multi-domain* will automatically be chosen for *projects* with *subprojects*, in which two or more *domains* are synchronized.
Application	This *customization characteristic* refers to the purpose for which a process is used..
Planning	A *project* that involves *model*ling *projects* as part of a *planning process*.
Design	A *project* that involves *model*ling *projects* for *design* purposes. A *design project* is typically preceded by a *planning project*, and typically puts higher requirements to *model* performance than the preceding *planning project*.
Operational management	A *project* that involves real-time operation, e.g. *flood forecasting* in water management.
Complexity	This *customization characteristic* refers to the comprehensiveness of the process and the commonness of using the process in projects (i.e. routine, research etc.).
Basic	A *project* is characterized as having a basic *complexity* when relatively small efforts and resources are required to carry out the *model*ling *project*. *Projects* in preliminary phases of a *planning project* aiming at a pre-screening of alternatives for subsequent more detailed *project* later on could e.g. be characterized as belonging to this category. The HarmoniQuA guidelines for basic job complexity is the simplest version of the guidelines, and it includes less tasks activities than e.g. the 'comprehensive' category.
Intermediate	A *project* is characterized as having an intermediate job complexity when relatively moderate efforts and resources are required to carry out the modelling *project*. The HarmoniQuA guidelines for intermediate job complexity is the medium version of the guidelines, as it includes more tasks and activities than the 'basic' category, but less than the 'comprehensive' category.
Comprehensive	A *project* is characterized as having a comprehensive job complexity when significant efforts and resources are required to carry out the modelling *project*. The HarmoniQuA guidelines for comprehensive job complexity are the most comprehensive version of the guidelines.

D.2 Ontological layer 2: modelling knowledge

Table D-3. The concepts of ontological layer 2 (modelling knowledge), All terms in italic are defined components of the meta-ontology (Chapter 5), or defined in the generic process ontology (Chapter 6, section 6.6.2 and in this Appendix section D.1).

Concepts	Meaning
Modelling paradigm	*Modelling* according to a model type and associated format, solver[1] and methodology. Note 1. *Modelling* is defined in the meta-ontology of Chapter 5. Note2. Synonym 'model solving paradigm'. Note 3. Next to *modelling paradigm* one can also distinguish *modelling* sub*paradigms* and / or *modelling* super*paradigms*. In the last case *simulation modelling* has to be seen as a *modelling* super*paradigm* and *continuous simulation* and *discrete simulation* as *modelling paradigms*. Note 4. *Modelling paradigm* will be restricted here to *mathematical modelling*. Note 5. Not included is 'business process modelling', as this is '(re)design of (business) processes'. Note 6. Not included is 'statistical modelling', as this is not including knowledge of processes governing the *problem* and *object system* at hand. These processes can belong various disciplines from biology, physics, sociology, economics, etc. Sometimes, statisticians distinguish data modelling (focused on data and some data fitting relation; example 'linear regression') and algorithmic modelling (focused on identifying an unknown, black box type of model; example 'neural nets'). Note 7. Not included is 'data modelling', which originates from the database community in computer science.
Selecting *model solving paradigm*	Select one or more appropriate *modelling paradigms* to use in *problem solving*.
Optimization *modelling paradigm*	The modelling paradigm, in which one seeks to minimize or maximize a real function by systematically choosing the values of real or integer variables from within an allowed set. Note 1. Adapted from Wikipedia. Note 2. Synonym: mathematical programming. Note 3. An *optimization* problem aims at optimizing (i.e. minimizing or maximizing) an *object function* in relation to mathematically formulated relations describing some functional relations in the object system
Linear programming *modelling paradigm*	Part of the *optimization modelling paradigm*, in which the objective function is linear and subject to linear equality and inequality constraints.
Non-linear programming *modelling paradigm*	Part of the *optimization modelling paradigm*, in which the objective function is non-linear or subject to non-linear equality and inequality constraints.
Simulation *modelling paradigm*	The modelling paradigm, which attempts to predict aspects of the behavior of some system by creating an approximate (mathematical) model of it. This can be done by physical modelling, by writing a special-purpose computer program or using a more general simulation package, probably still aimed at a particular kind of simulation (e.g. structural engineering, fluid dynamics). Typical examples are aircraft flight simulators or electronic circuit simulators. A great many simulation languages exist, e.g. Simula. Note 1. Adapted from Usenet newsgroup: news:comp.simulation.

[1] A 'solver' is a numerical algorithm to find solutions of a model. Each modelling paradigm has its own types of solvers.

Continuous *simulation modelling paradigm*	Part of the *simulation modelling paradigm*, which uses differential equations (either partial or ordinary). Periodically, the simulation program solves all the equations numerically, and uses the numbers to change the state and output of the simulation. Note 1. Adapted from: Wikipedia. Note 2. The system of differential equations represents (physical processes of the *object system*, in which the problem at hand is relevant. Note 3. 'Simulation program' is called here *site specific computer model*. Note 4. *Site specific computer model* is defined in Chapter 9 and Appendix G.
Discrete *simulation modelling paradigm*	Part of the *simulation modelling paradigm*, in which events are managed in time. In this type of simulation, the simulator maintains a queue of events sorted by the simulated time they should occur or the model checks at discrete time intervals if the state of has to be adapted. Note 1. Adapted from: Wikipedia. Note 2. Time advances occur with irregular increments in case of discrete event simulation models or with
Agent based *simulation modelling paradigm*	In agent-based *simulation*, the individual *entities* (such as molecules, cells, trees or consumers) in the model are represented directly (rather than by their density or concentration) and possess an internal state and set of behaviors or rules which determine how the agent's state is updated from one time-step to the next in interaction with other agents Note 1. Adapted from Wikipedia. Note 2. *Modelling* is defined in the meta-ontology of Chapter 5.
Modelling major *composite actions*	Groups of *modelling* related *composite actions*, including *basic modelling composite actions* and *modelling credibility composite actions*.
Basic modelling *composite actions*	*Composite actions* consisting of: • *Object system analysis*; • *Model* implementation • *Model* specification • Simulation
Object system analysis	*Composite action* defining the *object system* for the problem at hand and translating it in a *conceptual model*. Note 1. *Object system* is defined in Chapter 8 and Appendix F. Note 2. *Conceptual model* is defined in Chapter 9 and Appendix G. Note 3. *Object system analysis* can be compared with the term 'analysis' of Refsgaard and Henriksen (2004).
Model implementation	*Composite action* consisting of implementing the *conceptual model* into an appropriate computer program that can solve the mathematical model with some paradigm dependent solver, resulting in the computer model. Note 1. *Model implementation* can be compared with the term 'programming' of Refsgaard and Henriksen (2004).
Model specification	*Composite action* consisting of making the *computer model* site specific by adding the right *model inputs*. Note 1. *Model specification* can be compared with the term 'model set-up' of Refsgaard and Henriksen (2004). Note 2. Refsgaard and Henriksen (2004) state that 'model set-up' (here *model specification* is often associated with *model calibration*.
Model execution	*Composite action* consisting of running the *model* for its intended *model objective* and or running *model scenarios*. Note 1. *Model execution* can be compared with the term 'simulation' of Refsgaard and Henriksen (2004) as they focus on simulation models for water management. Note 2. *Model objective* and *model scenario* is defined in Chapter 9 and Appendix G.

Modelling credibility *composite actions*	*Composite actions* that evaluate the *model modes*: • *Model confirmation*: comparing *conceptual model* with *object system*; • *Computer model verification*: comparing *computer model* with *conceptual model*; • *Model calibration*: adjusting *site specific computer model* to *object system* according to scientific knowledge on processes in the *computer model*. • *Model validation*: comparing *site-specific computer model* with *object system*.
Model confirmation	Determination of adequacy of the *conceptual model* to provide an acceptable level of agreement with the *object system* for the *scope of conceptual model*. Note 1 Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) state that *model confirmation* is in other words the scientific confirmation of the theories/hypotheses included in the *conceptual model*. Note 3. *Conceptual model* is defined in Chapter 9 and Appendix G
Computer model verification	Substantiation that a *computer model* is in some sense a true representation of a *conceptual model* within certain specified limits or *scope of computer model* and corresponding ranges of accuracy. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) call this 'code verification'. Note 3. Refsgaard and Henriksen (2004) use 'ranges of application' instead of *scope of computer model*. Note 4.*Computer model* is defined in Chapter 9 and Appendix G. Note 5. The 'ranges of accuracy' is an *activity* of the *task* DETERMINE REQUIREMENTS in *ontological layer 3* (see section D.3)
Model calibration	The procedure of adjustment of *parameter* values of a *site specific computer model* to reproduce the response of the *object system* within the range of accuracy specified in the performance criteria. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) use 'reality' instead of *object system*. Note 3. The 'ranges of accuracy' and 'performance criteria' are *activities* of the *task* DETERMINE REQUIREMENTS in *ontological layer 3* (see section D.3)
Model validation	Substantiation that a *site specific computer model* within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the *site specific computer model*. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) use 'model' instead of *site specific computer model*. Note 3. Refsgaard and Henriksen (2004) use 'domain of applicability' instead of *scope of site specific computer model*. Note 4. The 'ranges of accuracy' is an *activity* of the *task* DETERMINE REQUIREMENTS in *ontological layer 3* (see section D.3)

D.3 Ontological layer 3: simulation modelling knowledge

In Chapter 6 a process ontology for multidisciplinary model-based problem-solving has been proposed, which is structured in so called *ontological layers*, which aim to facilitate reuse of parts of the ontology. *Ontological layer 3* is an ontological specialization of the process ontology containing simulation modelling knowledge. In this section all concepts used in that ontology are defined. It contains all concepts depicted in the structure diagram of Figure 6-9 of Chapter 6.

Steps will be presented with *name* and *description*, *tasks* by presenting their *names*, *definition* and *explanation* (if available). All other *task elements*, including *activities* and *methods*, are left out here, because presenting them here would request several hundreds of pages. The complete textual version of the guidelines can be found on www.HarmoniQuA.org/public/Product/software.htm.

Table D-4.Instances of the concepts steps and tasks of ontological layer 2, the specialized process ontology for modelling, are presented by name and definition. The further task properties can be found on the accompanying CD-ROM or on http://harmoniqua.org/public/Product/software.htm. Steps are left aligned and bold. Tasks are right aligned and small caps. All task definitions are from the HarmoniQuA Guidelines, version 2.3.

Concepts	Meaning
Model Study Plan	This step aims at setting-up a project by performing the following tasks: • Describe Problem and Context • Define Objectives • Identify Data Availability • Determine Requirements • Prepare Terms of reference • Proposal and Tendering • Agree on Model Study Plan and Budget
DESCRIBE PROBLEM AND CONTEXT	*Task* definition of the *step Model Study Plan*: A clear, precise (not necessarily quantitative) specification of the known problem details, the context of the study and the type of calculations to be made.
DEFINE OBJECTIVES	*Task* definition of the *step Model Study Plan*: Specification of the goals to be achieved.
IDENTIFY DATA AVAILABILITY	*Task* definition of the *step Model Study Plan*: Make a structured list of data, including its relevance, availability and quality for the problem at hand.
DETERMINE REQUIREMENTS	*Task* definition of the *step Model Study Plan*: Specify requirements to the quality of the model study.
PREPARE TERMS OF REFERENCE	*Task* definition of the *step Model Study Plan*: Terms of Reference is the specifications for the modelling job to be carried out.
PROPOSAL AND TENDERING	*Task* definition of the *step Model Study Plan*: Tendering is the process of selecting a consultant for a modelling job on the basis of technical and financial proposal(s).
AGREE ON MODEL STUDY PLAN AND BUDGET	*Task* definition of the *step Model Study Plan*: The water manager and the modeler reach an agreement on the technical and financial conditions of the modelling job.
Data and Conceptualization	This *step* aims at • An adequate description of the system • Correct handling and evaluation of data • An adequate description • Evaluation of conceptual models • Selecting (existing) model code by performing the following *tasks*: • Describe System and Data Availability • Process Raw Data • Sufficient data? • Model Structure and processes • Model Parameters • Summarize Conceptual Model and Assumptions • Need for Alternative Conceptual Models? • Process Model Structure Data • Assess Soundness of Conceptualization • Code Selection • Report and Revisit Model Study Plan (Data and Conceptualization • Review Data and Conceptualization and Model Set-up Plan

DESCRIBE SYSTEM AND DATA AVAILABILITY	*Task* definition of the *step Data and Conceptualization*: Identify the processes to be represented, and the data needed, to setup and validate a model of sufficient complexity to meet the objectives defined in the Model Study Plan.
PROCESS RAW DATA	*Task* definition of the *step Data and Conceptualization*: Collection, pre-processing and evaluation of raw data.
SUFFICIENT DATA?	*Task* definition of the *step Data and Conceptualization*: Determine if there is sufficient qualified data to fulfill the project objectives as defined in the Model Study Plan. Qualified data refers to data that has been collected, evaluated and deemed applicable for use in the model study.
MODEL STRUCTURE AND PROCESSES	*Task* definition of the *step Data and Conceptualization*: Delineation of the physical extent, dimensionality, internal framework, boundaries and processes of the model domain, and a description of how they will be represented mathematically.
MODEL PARAMETERS	*Task* definition of the *step Data and Conceptualization*: Framing of parametric representation or parameterization of system characteristics.
SUMMARIZE CONCEPTUAL MODEL AND ASSUMPTIONS	*Task* definition of the *step Data and Conceptualization*: Summarize the current understanding of the system in terms of verbal descriptions, graphical presentations, equations, governing relationships, and/or natural laws that purport to describe reality
NEED FOR ALTERNATIVE CONCEPTUAL MODELS?	*Task* definition of the *step Data and Conceptualization*: Review conceptual model to determine whether there is sufficient uncertainty over the model structure and processes to warrant the development and evaluation of alternative formulations.
PROCESS MODEL STRUCTURE DATA	*Task* definition of the *step Data and Conceptualization*: Transform and/or organize the qualified data into a database for use in constructing the model.
ASSESS SOUNDNESS OF CONCEPTUALIZATION	*Task* definition of the *step Data and Conceptualization*: An evaluation of the credibility and suitability of the conceptual model based upon professional judgment.
CODE SELECTION	*Task* definition of the *step Data and Conceptualization*: Decide on appropriate modelling software that is consistent with the conceptual model and the study objectives.
REPORT AND REVISIT MODEL STUDY PLAN (DATA AND CONCEPTUALIZATION	*Task* definition of the *step Data and Conceptualization*: Report on step Data and Conceptualization, to cover the database and metadata, and also the conceptual model and its soundness. Draft associated Chapters of the Project Report. Revise the Model Study Plan with respect to the tasks in the next modelling step.
REVIEW DATA AND CONCEPTUALIZATION AND MODEL SET-UP PLAN	*Task* definition of the *step Data and Conceptualization*: Review the work carried out during the modelling step Data and Conceptualization.
Model Set-up	This *step* aims at setting-up the model to be used by performing the following *tasks*: • Construct Model • Test Runs Completed • Specify and Update Calibration and Validation Targets and Criteria • Report and Revisit Model Study Plan (model Set-up) • Review Model Set-up and Calibration and Validation Plan
CONSTRUCT MODEL	*Task* definition of the *step Model Set-up*: Construct model is the process of transforming a conceptual model into a model set-up for the study area using the selected model code, where both set-up data and simulations parameters have been defined.

TEST RUNS COMPLETED	*Task* definition of the *step Model Set-up*: A test run is carried out in order to identify any problems with the constructed model (set-up data and computational control parameters) and/or execution of the selected code.
SPECIFY AND UPDATE CALIBRATION AND VALIDATION TARGETS AND CRITERIA	*Task* definition of the *step Model Set-up*: Calibration and validation accuracy targets should be proposed as measures for the acceptance criteria prior to undertaking model calibration.
REPORT AND REVISIT MODEL STUDY PLAN (MODEL SET-UP)	*Task* definition of the *step Model Set-up*: Report the Model set-up and write the associated Chapter of the Project Report and revising the Model Study Plan with respect to the tasks in the next modelling step.
REVIEW MODEL SET-UP AND CALIBRATION AND VALIDATION PLAN	*Task* definition of the *step Model Set-up*: Review the work carried out during the modelling step Model set-up.
Calibration and Validation	This *step* aims at calibrating and validating the model to be used by performing the following *tasks*: • Specify Stages in Calibration Strategy • Select Optimization Method • Define Stop Criteria • Select Calibration parameters • Parameter Optimization • All Calibration Stages Completed? • Asses Soundness of Calibration • Validation • Asses Soundness of Validation • Uncertainty Analysis of Calibration and Validation • Scope of Applicability • Report and Revisit Model Study Plan (Calibration and Validation) • Review Calibration and Validation and Simulation Plan
SPECIFY STAGES IN CALIBRATION STRATEGY	*Task* definition of the *step Calibration and Validation*: Subdivision of the entire calibration process into sub-steps.
SELECT OPTIMIZATION METHOD	*Task* definition of the *step Calibration and Validation*: Select a parameter optimization method for model calibration
DEFINE STOP CRITERIA	*Task* definition of the *step Calibration and Validation*: Define one or more criteria to determine when to stop calibration.
SELECT CALIBRATION PARAMETERS	*Task* definition of the *step Calibration and Validation*: Determine which parameters (and other model input, e.g. decision variables) will be used for calibration and which will be kept constant for all calibration model runs.
PARAMETER OPTIMIZATION	*Task* definition of the *step Calibration and Validation*: Execute parameter optimization.
ALL CALIBRATION STAGES COMPLETED?	*Task* definition of the *step Calibration and Validation*: To evaluate whether all steps in calibration strategy have been finalized satisfactorily.
ASSES SOUNDNESS OF CALIBRATION	*Task* definition of the *step Calibration and Validation*: The assessment of the soundness of the calibration is an evaluation of the credibility of the calibrated model based upon professional judgment.
VALIDATION	*Task* definition of the *step Calibration and Validation*: Substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model.

ASSES SOUNDNESS OF VALIDATION	*Task* definition of the *step Calibration and Validation*: The assessment of the soundness of the validation is an evaluation of the credibility of the calibrated model based upon professional judgment.
UNCERTAINTY ANALYSIS OF CALIBRATION AND VALIDATION	*Task* definition of the *step Calibration and Validation*: Quantification of uncertainty in the calibrated and validated model due to incomplete knowledge of model parameters, input data, boundary conditions and conceptual model. In an uncertainty analysis the combined effects of these uncertainties are taken into account.
SCOPE OF APPLICABILITY	*Task* definition of the *step Calibration and Validation*: This task describes the circumstances or conditions under which the model has documented predictive capabilities.
REPORT AND REVISIT MODEL STUDY PLAN (CALIBRATION AND VALIDATION)	*Task* definition of the *step Calibration and Validation*: Reporting the model calibration and validation and writing the associated Chapters of the Project Report and revising the Model Study Plan with respect to the tasks in the next modelling step.
REVIEW CALIBRATION AND VALIDATION AND SIMULATION PLAN	*Task* definition of the *step Calibration and Validation*: Review of the tasks carried out within the modelling step Calibration and validation.
Simulation and Evaluation	This *step* aims at defining and running the scenarios that will be used to solve the problem at hand by performing the following *tasks*: • Set-up Scenario • Simulations • Check Simulations • Analyze and Interpret results • Assess Soundness of Simulation • Uncertainty Analysis of Simulation • All Scenarios Completed • Reporting of Simulation and Evaluation • Review of Simulation and Evaluation • Need for Post Audit • Model Study Closure
SET-UP SCENARIO	*Task* definition of the *step Simulation and Evaluation*: Adjust validated model to match the scenario, and prepare any time-series data needed to run the model.
SIMULATIONS	*Task* definition of the *step Simulation and Evaluation*: Use of a validated model to quantify the response of the system to possible future events. Real-time runs are aimed at producing model output to estimate future states of the system within the lead time of the forecast. Simulations are made by running the model with a set of validated parameters and imposing expected future stresses and/or management scenarios.
CHECK SIMULATIONS	*Task* definition of the *step Simulation and Evaluation*: Screening the results of the simulations for possible errors. The check of simulations is an inspection of the model results with the objective of exposing extreme or incorrect model output.
ANALYZE AND INTERPRET RESULTS	*Task* definition of the *step Simulation and Evaluation*: Critical examination of the results of the simulations.
ASSESS SOUNDNESS OF SIMULATION	*Task* definition of the *step Simulation and Evaluation*: A qualitative assessment of the model simulations that takes into account quantitative measures and practical experience.

UNCERTAINTY ANALYSIS OF SIMULATION	*Task* definition of the *step Simulation and Evaluation*: Assessment of uncertainty in model results due to incomplete knowledge of model parameters, input data, boundary conditions and conceptual model. Furthermore the uncertainty originating from the decision context (external factors) may be included.
ALL SCENARIOS COMPLETED	*Task* definition of the *step Simulation and Evaluation*: Decide whether another scenario needs to be analysed. If not, compare results from those already analysed.
REPORTING OF SIMULATION AND EVALUATION	*Task* definition of the *step Simulation and Evaluation*: Reporting of the results from the modelling step Simulation and Evaluation.
REVIEW OF SIMULATION AND EVALUATION	*Task* definition of the *step Simulation and Evaluation*: Review of the tasks carried out within the modelling step Simulation and Evaluation.
NEED FOR POST AUDIT	*Task* definition of the *step Simulation and Evaluation*: A post audit (or a post project appraisal) is a review of the modelling study carried out after some time when new information is available.
MODEL STUDY CLOSURE	*Task definition* of the *step Simulation and Evaluation*: Formal completion of the modelling job.

D.4 Ontological layer 4: simulation modelling knowledge for water management

In Chapter 6 a process ontology for multidisciplinary model-based problem-solving has been proposed, which is structured in so called *ontological layers*, which aim to facilitate reuse of parts of the ontology. *Ontological layer 4* contains simulation modelling knowledge for water management.

Table D-5 contains specializations of the water management *subdomains*. As are depicted in the structure diagram of Figure 6-10 in Chapter 6.

Table D-5. Concepts of ontological layer 4 (simulation modelling knowledge for water management) specializing water management subdomains, i.e. , All terms in italic are defined components of other ontological layers. The definitions are derived from the HarmoniQuA Glossary.

(Modelling for Water Management) Domain	This *customization characteristic* 'domain' refers to the scientific *discipline* of water management, to which the *model*ling work for water management belongs.
Hydrodynamics	The hydrodynamic modelling for water management domain comprises flows of water, sediment transport and morphological aspects in all surface water elements such as: • Water supply systems • Sewer drainage systems • River systems including flood plain, lakes and reservoirs • Estuaries and coastal waters The hydrodynamic for water management domain in HarmoniQuA corresponds to the water flow and sediment aspects in 'river' 'lake' 'transitional water' and 'coastal water' as these terms are defined in the Water Framework Directive.

Groundwater	The groundwater modelling for water management *domain* (quantity and quality): comprises *model*ling activities related to *groundwater*, i.e.: • *Groundwater* flow (heads and flows) • Solute transport (advection, diffusion and hydrodynamic dispersion) • Geochemistry (ad/absorption, ion exchange, complexation, degradation, etc.) The *groundwater domain* in HarmoniQuA corresponds to the term 'groundwater' as this term is defined in the Water Framework Directive
Precipitation-Runoff	The *precipitation-runoff domain* comprises *model*ling *activities* related to *precipitation-runoff* processes, i.e.: • Flow processes (precipitation, evapotranspiration, runoff etc., typically at catchment scale) • Non-point pollution (e.g. nitrate, pesticides originating from agricultural practice) The *precipitation-runoff domain* in HarmoniQuA corresponds more or less to the term 'surface water' related to 'river basin' as these terms are defined in the Water Framework Directive.
Flood forecasting	The flood forecasting modelling for water management domain comprises all modelling activities related to operational, real-time forecasting of floods in rivers and coastal waters. The flood-forecasting domain in HarmoniQuA is applicable to the 'river', 'transitional water' and 'coastal water' as these terms are defined in the Water Framework Directive.
Surface water quality	The surface water quality modelling for water management domain comprises water quality processes in all surface water elements such as: • River systems including flood plain, lakes and reservoirs • Estuaries and coastal waters The surface water quality domain in HarmoniQuA corresponds to the water quality aspects in 'river' 'lake' 'transitional water' and 'coastal water' as these terms are defined in the Water Framework Directive.
Biota / ecology	The biota (ecological) modelling for water management domain comprises water quality processes in all surface water elements such as: • River systems including flood plain, lakes and reservoirs • Estuaries and coastal waters The biota (ecological) domain in HarmoniQuA corresponds to the ecological aspects in 'river' 'lake' 'transitional water' and 'coastal water' as these terms are defined in the Water Framework Directive.
Socio-economics	The socio-economic modelling for water management domain comprises all modelling activities related to socio-economic aspects.

The content of *tasks* in *simulation modelling* can be specialized for water management in several ways. In all cases the specialized simulation modelling knowledge, specific for water management is *instantiated* by *activities* that are specific for specific water management *subdomains*. This can be done by defining extra *activities* for one or more *subdomains*. An example of this approach is illustrated in Table D-6 which contains the activities for the task ‘Summarize Conceptual Model and Assumptions’. In another approach some activities have alternative activities for one or more domains, as is illustrated in Table D-7 for the task ‘Specify or Update Calibration and Validation Targets and Criteria’.

Table D-6. Subdomain specific activities of the task 'Summarize Conceptual Model and Assumptions' (task 2.6) as an example of concepts in ontological layer 4 (simulation modelling knowledge for water management). All activity descriptions are derived from the HarmoniQuA Guideline. GE = generic domain (from ontological level 1), MD = multi-domain, BI = biota (including ecology), FF = flood forecasting, GW = groundwater (both water quantity and water quality, HD = hydrodynamics, PR = precipitation-runoff, SE = socio-economics, WQ = surface water quality. MO = modeller (role).

Subdomain specific activities of the task 'Summarize Conceptual Model and Assumptions'			
Activity name	Subdomain	Role	Description
Summarize conceptual model	BI, FF, GE, PR, WQ, HD	MO	The conceptual model should be summarized using graphical presentations, descriptive text and equations
Model boundaries	GW	MO	Describe the location and type of boundaries for the model area. Boundary types include specified flow, specified head and head-dependent flow. If transport is included, the boundary for the included species must also be included, e.g. source strength and temporal/spatial variation.
Geological framework	GW	MO	Describe the geological units and corresponding hydrostratigraphic units, model layers and associated aquifer properties.
Hydrological framework and stresses	GW	MO	Describe the recharge and discharge processes and dominant flow mechanisms. This includes the definition of the aquifer media type (porous medium, fractured, etc.) and groundwater-surface water interaction. Groundwater level measurements should be used to estimate the general direction of groundwater, the location of discharge and recharge areas and the connection between groundwater and surface water flow.
Human-induced factors	GW	MO	Describe the location and type of human-induced factors such as drains pumping wells and sources of pollutions.
Water budget	GW	MO	A water budget should be prepared from the field data to summarize the magnitude of inflows, outflow and changes in storage. See also activity: Water budget in task *Model Structure and Processes*
Socio-economic assessment criteria	SE	MO	Summarise the socio-economic assessment criteria applied in the model and their assumptions. Examples are the parameters used regarding costs (e.g. which calculation method?), economic benefits (opportunity cost or willingness-to-pay approach?), least-cost and /or cost benefit analysis (which rate of interest?) and multi-criteria analysis (which weighting factors?).
Linking hydrologic with socio-economic model components	SE	MO	In summarising the linkages of hydrological with socio-economic model components consider the various types of coupling. Principally it can be differentiated between the holistic and modular approach. In the first case, hydrologic (including engineering) components are directly coupled with socio-economic variables and parameters. Those models may be become very complex. In the second case, separate socio-economic modules exist (e.g. cost minimisation tool) which are linked with the relevant components of the water model. The advantage of this approach is that the hydrological components must not be simplified or changed in order to tune them to socio-economic components. However, mixed types of approaches are also possible.

List Assumptions	BI, GE, PR, HD, FF, WQ	MO	Prepare a comprehensive list of assumptions made during development of the following elements of the conceptual model: • Model structure • Description of processes • Parameterisation, including fixed parameter values and likely ranges for calibration parameters
Hydrogeological Framework Assumptions	GW	MO	Develop a comprehensive list of assumptions made in defining the following: • Extent of model area • Hydrostratigraphic units • Boundary conditions
Processes Assumptions	GW	MO	Develop a comprehensive list of assumptions made in defining how the following processes are represented in the model: • Recharge processes • Unsaturated zone processes • Groundwater/surface water interaction • Preferential flow paths (macropores, fractures and/or fault zones) • Discharge processes • Solute transport and water quality
Parameterization Assumptions	GW	MO	Develop a comprehensive list of the assumptions made in defining the following: • Parameters that are specified as constants, including fixed parameter values • Parameters that will be estimated by model calibration, including the likely range of fitted values.
Human response assumptions	SE	MO	Develop a list of the assumptions made with regard to the modelling of human response functions.
Human-induced factors	PR, FF	MO	Describe the location and type of human-induced factors such as dams, weirs, water abstractions that may have a significant impact on modelling. In case of diffuse pollution modelling, describe the non-natural sources of pollutant that may play a role in the modelling.

Table D-7. Subdomain specific alternative activities of the task 'Specify or Update Calibration and Validation Targets and Criteria' as an example of concepts in ontological layer 4 (simulation modelling knowledge for water management). All activity descriptions are derived from the HarmoniQuA Guideline . GE = generic domain (from ontological level 1), MD = multi-domain, BI = biota (including ecology), FF = flood forecasting, GW = groundwater (both water quantity and water quality, HD = hydrodynamics, PR = precipitation-runoff, SE = socio-economics, WQ = surface water quality. MO = modeller (role).

Subdomain specific alternative activities of the task 'Specify or Update Calibration and Validation Targets and Criteria'			
Activity name	Subdomain	Role	Description
Select observation datasets	BI, GE, WQ, HD	MO	Selection of appropriate calibration datasets for each step in calibration (steady state, transient, etc.) is dependent on the quantity and quality of the available data. For steady state conditions the calibration data should be as representative of "average flux and stage conditions" as possible. For transient conditions it may be necessary to consider temporal representativeness (e.g. daily values may not be relevant for monthly stress period, but should be averaged).

Select observation datasets	GW	MO	Calibration data sets should be selected based upon the modelling objectives and quantity and quality of the available data. Models used to predict changes in groundwater levels, capture zones, changes in groundwater/surface water interaction and other boundary fluxes should be calibrated to both hydraulic head and flux data. Moisture content data are desirable if simulations include the unsaturated zone. Calibration data for models used to predict solute transport should include groundwater ages and/or water quality data. For steady state simulations the calibration data should be representative of average head and flow conditions. For transient conditions it may be necessary to consider temporal representativeness (e.g. daily values may not be relevant for monthly stress period, but should be averaged).
Select observation datasets	PR	MO	Selection of appropriate calibration datasets for calibration is dependent on the quantity and quality of the available data. It may be necessary to consider temporal representativeness (e.g. daily values may not be relevant for monthly stress period, but should be averaged). Streamflow (and pollutant concentration for diffuse pollution study) datasets for calibration should ideally include a wide range of flow events (low to high).
Select observation datasets	FF	MO	Selection of appropriate calibration datasets for calibration is dependent on the quantity and quality of the available data. In case of real time applications calibration datasets may contain any kind of observation data while datasets for validation should be derived from real time data. In case of models with a strong physical background more accurate and detailed datasets for calibration decrease uncertainty of model parameters. It may be necessary to consider temporal representativeness (e.g. hourly values reflecting diurnal changes may be averaged for days). Streamflow datasets for calibration should ideally include a wide range of events from low to high flows. For event-based modelling, a statistically significant sample of events should be available. In case of clustering flood events each cluster/category should contain a significant number of events. Datasets for extreme events should have special values.
Select observation datasets	SE	MO	Selection of appropriate calibration datasets for each step in calibration (steady state, transient, etc.) is dependent on the quantity and quality of the available data. Models used to predict, for instance, changes in water demand of different water users, wastewater disposal of different point sources, etc. should be calibrated to water supply (or water availability), assigned water effluent rights, charges on water services (e.g. water supply) etc. For steady state conditions the calibration data should be as representative of e.g. average water balance, compliance with ambient water quality objectives, cost/benefit ratios of different water uses (e.g. groundwater abstraction), etc. as possible. For transient conditions it may be necessary to consider temporal representativeness (e.g. daily / monthly values may not be relevant for monthly / yearly stress periods, but should be averaged).

D.5 References

Refsgaard, J.C. and H.J. Henriksen, 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.

Appendix E Requirements to the Modelling Support Tool, MoST

Content of Appendix E, belonging to chapter 7

E.1 Introduction

In chapter 7 the requirements to the Modelling Support Tool, MoST, have been discussed. Here more details are given on the requirement analysis method and criteria classification (section E.2) and on how the criteria are used in the implementation of MoST.

E.2 Requirement analysis method

The guidelines generated from the content of the process knowledge base discussed in the previous chapter, has a value in itself, but to use it a Modelling Support Tool is needed that uses the ontological guidelines, guide the team through the network of things to do and keeps records of what is actually done. This section will outline the requirements to such a tool. Just as in the previous chapter, the requirements will be classified according to the classification of Glinz (2005), here summarized as follows (**bold** are facets of the requirements and *italic* are values of facets):

1. **Kind**: *function*, *data*, *performance*, *specific quality*, *constraint*;
2. **Representation**: *operational*, *quantitative*, *qualitative*, *declarative*;
3. **Satisfaction**: *hard*, *soft*;
4. **Role**: *prescriptive*, *normative*, *assumptive*.

These values / facets are explained in Appendix C.

E.3 Design requirements to MoST

Table E-1 outlines the requirements and the associated consequences for the design of the Process Support Tool and how each requirement fits in Glinz' classification.

Table E-1 summarizes the requirements to a Modelling Support Tool, which matches the mental model for process support (including modelling), as introduced and updated in Chapter 6 and Chapter 7. Chapter 7, section 7.4 discusses the Modelling Support Tool as implemented for HarmoniQuA, based on the requirements of Table E-1.

Table E-1. Requirements to the Modelling Support Tool, categorized according to Glinz (2005).

Requirement	Design solution	Categories
1. Modelling Support Tool[1] (MoST) should support multidisciplinary teams during the whole project lifecycle.	This support concerns: 1. Initializing projects; 2. Providing guidance (from the KB); 3. Monitoring what is actually done; 4. Help to generate reports for various audiences and purposes.	• Function • Operational • Hard • Prescriptive
2. A GUI Menu should give access to all (most) functionality.	The menu includes the following menu items, each with a dropdown menu: • **File** o *New Online Project* o *New Local Project* o *New Project From Templates* o *Open Online Project* o *Open Local Project* o *Save Local Project /* o *Save Local Project As …* o *Save The Project As Template* o *Close Project*	• Function • Operational • Hard • Prescriptive

[1] In fact, MoST is a generic Modelling Support Tool, but here associated with the KB for model-based water management. Furthermore, MoST is also used for AquaStress in a slightly improved version.

- o *List With Recently Opened Projects*
- o *Exit*

- **Guideline**
  - o *Filter*
  - o *Find*
  - o *Find Next*
  - o *Tree*
    - ▪ *Expand All*
    - ▪ *Collapse All*
    - ▪ *Small Icons*
    - ▪ *Medium Icons*
    - ▪ *Large Icons*
  - o *Flowchart*
    - ▪ *Zoom In*
    - ▪ *Zoom Out*
    - ▪ *Fixed Font Size*
    - ▪ *Select Font*
  - o *Task View*
    - ▪ *Save Active Panel To HTML*
    - ▪ *Print // Select Guidelines*
    - ▪ *Download Guidelines*
  - o *Print*
  - o *Select Guideline*
  - o *Download Guideline*
- **Project**
  - o *Convert Local Project To Online*
  - o *Export Online Project To Local*
  - o *Open Scoreboard*
    - ▪ *Subproject 1*
    - ▪ *Subproject 2*
    - ▪ *Etc.*
  - o *Confirm Open Task*
  - o *Confirm Start Activity*
  - o *Confirm Decisions*
  - o *Logon As Different User*
  - o *Change Password /*
  - o *Tree*
    - ▪ *Expand All*
    - ▪ *Collapse All /*
    - ▪ *Small Icons*
    - ▪ *Medium Icons*
    - ▪ *Large Icons*

	o *Project Settings* • **Options** o *Show Toolbar* o *Show Text On Toolbar Buttons* o *Show Welcome Screen On Startup* o *Server Settings* o *File Settings* o *Edit Colors* • **Help**: o *Online Help* o *Glossary* o *Highlight Glossary Terms* o *Disclaimer* o *Credits* o *About* Many menu items will be discussed in Chapter 7, section 7.4.	
3. MoST should present the guidelines from the KB in various ways in order to support its use efficiently and effectively.	This is realized by the guideline component, which provides three views on the KB: a. **Tree view**: resembles standard windows trees like and allow easy browsing through *steps* and *tasks*; b. **Flowchart view**: allows browsing through the network of *steps* and *tasks*, but provides also a structured view with the order of *tasks* and feedback loops. c. **Task view**: is a detailed, textual description of a task.	• Function • Operational • Hard • Prescriptive
4. MoST should enable one to set-up projects.	Setting –up projects includes: give project name and indicate version, define subprojects and associated domains / disciplines, select relevant tasks per subproject, define team members, specify roles per team member, authorize them per subproject and role and appoint project administrator(s) and edit the scoreboard template questions.	• Function • Operational • Hard • Prescriptive
5. MoST should monitor what team members do.	The project execution component has three views: a. **Tree view**: displays all the tasks in the project and the status of these tasks. This view can be used to browse through the necessary tasks and also to inspect the status of each task in the subproject. If the prescribed order in the guidelines is enforced, tasks have to be performed in the right order. b. **Task view**: The task view provides guidance on the task that is selected in the tree view, with tabs to select the sections on the guidance for that task, just as in the guidance component. c. **Activity view**: is the main panel for recording what team members do in a process journal (e.g. in case of modelling a model journal). It should includes the following functionalities: • Starting, skipping and completing tasks;	• Function • Operational • Hard • Prescriptive

	• Recording start date, end date, (time) resources spent, etc.; • Filtering activities per role and domain; • Starting and completing activities; • Describing details, actions and outcomes of activities; • Selecting methods used; • Attach documents and retrieve and delete attached documents (of various types) to the model journal; • In case of decision tasks and review tasks, decide which is the next task to do.	
6. MoST should help making reports.	Reports will be generated from the content of a process (model) journal, according to two criteria: its purpose and intended audience. *Purposes* can be: • Informing other team members, • Management checks on progress, • The final report for the client of the study done in the project. • Etc. Audiences are also very diversified: • Scientists; • Managers; • Professional engineers; • (Lay) stakeholders; • Interested members of the general public. All content of reports has to be generated from a journal, so most adaptations for usage and audience are in the selection of the information stored in the journal, limiting the flexibility of report content.	• Function • Operational • Hard • Prescriptive
7. Glossary with terminology (per discipline) should be included.	The glossary should contain all content terminology, for model-based water management or for some other application domain.	• Function • Operational • Soft • Normative
8. The GUI of the tool should be easy to use without a steep learning curve, but it should also include most options wished by professionals that have to work with it on a daily basis.	This requirement is rather contradictory. The only solution is sufficient help and training material for various audiences.	• Specific quality • Declarative • Soft • Prescriptive
9. The Modelling Support Tool should be portable to other operating systems (platforms).	MS Windows™ is the dominant platform, but a substantial part of modelers uses other platforms, most Unix (or its variants). The Modelling Support Tool and the server application are developed in Java. This guarantees platform independency.	• Specific quality • Constraint • Hard • Normative

10. The components of the tool (providing guidance, project initialization, project running, reporting) should be reliable, multipurpose, easy to maintain/update and fast..	Reliability is most required for the functionality that supports teamwork; what people did should be carefully recorded and filed; multipurpose features are needed for extensive use by professionals; easy maintenance and updates are obvious for the knowledge base, but also the tool should be adaptable to new insights; the size of the KB requires that the tool should be able to fast present the relevant parts of the KB.	• Performance • Qualitative • Soft • Normative
11. The Modelling Support Tool should automatically use changes in the KB.	The Modelling Support Tool should be capable of changes in the KB. The KB is not a static monolithic body of knowledge, but a flexible representation of the state-of-the-art in some problem-solving field, e.g. modelling for water management. The tools should deal with new content of the KB and preferably also handle changes in the structure of the KB correctly or, at least, the tool should be adapted to the new KB structure easily. In the other direction the tool generates and edits journals, which have the same structure as the KB, but with some extensions for information on what team members fill-in in the journal and references to attached files. This requires a well-defined interface between the KB, the tool and the journals. This has been realized by using the ontological structure for an XML definition, which is been used by Protégé2000, the client tool and the server application.	• Constraint • Operational • Hard • Normative
12. The Modelling Support Tool should be designed in such a way that it facilitates adapting it for other languages.	In some countries process guidance in English will not obstruct its use by professionals. Other countries and other target groups will quite certain be hindered if English is the only language supported in the tool and the KB. Translating the KB is straightforward and will require limited resources, while adapting the tool for other languages requires a proper design, in which all language elements are separated from the rest of the code. With such a design translation of the tool is easy and costs hardly resources.	• Constraint • Declarative • Hard • Prescriptive
13. The Modelling Support Tool should be used by a distributed team working in a network setting (LAN or Internet).	In a cooperation project team members have typically their own computer(s) and each project, in which a team cooperates, has a single journal. This requires a client-server architecture. There should be a *KB server*, a *project server* and *client computers* for team member.	• Function • Qualitative • Hard • Assumptive
14. The components of the system should be distributed in an efficient and effective network structure.	The criteria for distributing the components over the client-server-architecture: • KB at a single server; • Project journals accessible for teams (preferably LAN or protected on Internet; • Clients (desktop PC's, notebooks) with CPU intensive tasks.	• Performance • Declarative • Soft • Normative
15. A single central server should be used for all versions of the KB in order	The KB should be made accessible through the Internet at a single and central site in order to allow easy maintenance and upgrading.	• Specific quality • Operational • Hard • Normative

to facilitate maintenance and upgrading.	On this site a server application should run, which does the server side of what has to be done, e.g. providing the KB is some client requests. The KB editor is a web application allowing all authorized persons to edit the KB (including the glossary) and all other registered persons to add comment.	
16. Other servers should be used for projects.	Project journals should be stored on the project server (LAN or Internet). The server application should synchronize work of individual team members to a reliable project journal containing the work of the whole team.	• Specific quality • Operational • Hard • Normative
17. Computers with connection to the central server and to the project server should run a client application for carrying out project work per team member.	Team member should work with the client application MoST. MoST communicates also with the server, including requests to the server (e.g. provide a specific version of the KB, send automatically the project journal of a team member to the server and change the shared project journal on the server accordingly.	• Specific quality • Operational • Hard • Normative

The overall structure of the Process Support System (tool and KB) is depicted in Figure E-1, including a central server, a project server and a client computer.

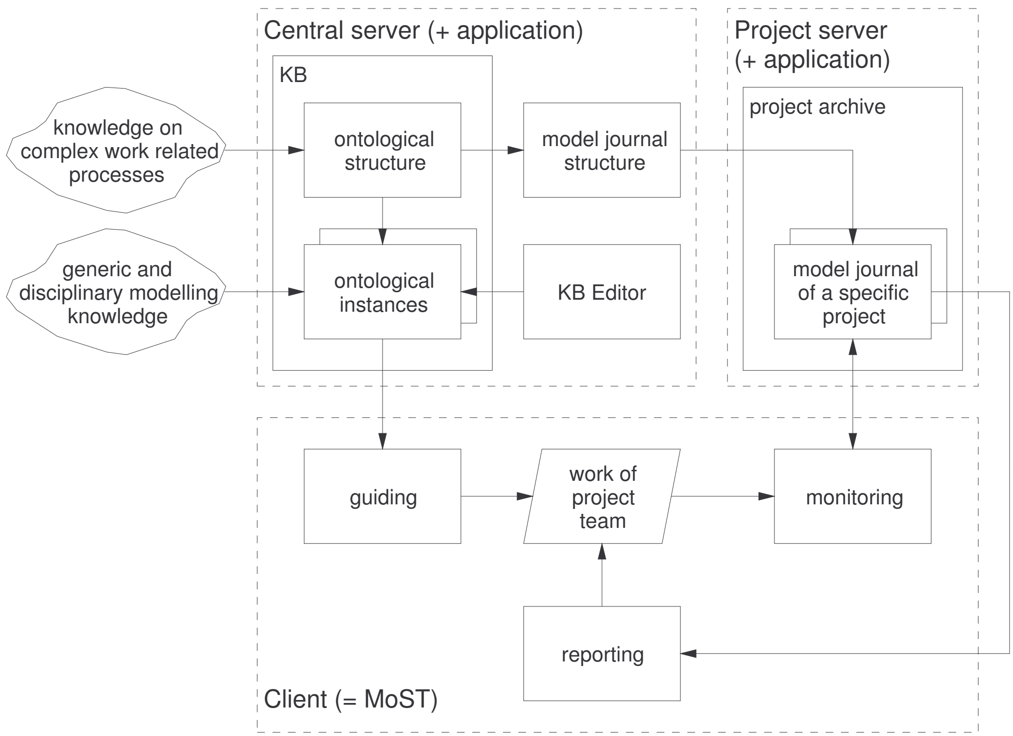


Figure E-1. Sketch of the client-server architecture with MoST's process knowledge base and its major functions, being initializing projects, guiding, monitoring and reporting. The process guidelines are stored in the ontological instances. Journals of specific projects are located on a project server, being a LAN (team projects, in which teams are within a single organization) or some controlled Internet server (team project with distributed team members). Adapted from Scholten et al., 2006.

E.4 References

Glinz, M., 2005, Rethinking the Notion of Non-Functional Requirements, Proceedings of the Third World Congress for Software Quality (3WCSQ 2005), Munich (D), 55-64, http://www.ifi.unizh.ch/groups/req/ftp/papers/3WCSQ2005.pdf.

Appendix F Concepts of the problem and OS ontology

Content of Appendix F, belonging to chapter 8

F.1 Ontological layer 1: generic problem knowledge

In Chapter 8 a problem and object system ontology has been proposed, which is structured in so called *ontological layers* aiming to facilitate reuse of parts of the ontology. *Ontological layer 1* contains generic problem knowledge. In this Appendix all concepts used in that ontology are defined in Table F-1. It contains all concepts depicted in the structure diagram of Figure 8-3 of Chapter 8.

Table F-1. Generic problem ontology concepts (*ontological layer 1*). All terms in *italic* are defined components of the meta-ontology (Chapter 5).

Concepts	Meaning
Object system boundary	Demarcation between *object system entities* that belong to the *object system* and what does not belong to the *object system*. Note1. *Object system relations* between *object system entities* are also within the *system boundary*, but some *object system entities*. Note 2. *Object system* is defined in the meta-ontology of Chapter 5.
Object system environment	*Relevant entities* that do not belong to the *object system*, but influence *object system entities*.
Object system context	The aspects and situation, which makes an *object system* relevant for the *problem*.
Object system aggregation level	The level of detailing the *object system*.
Object system structure	The content of the *object system*, consisting of chosen *object system entities* and *object system relations*.
Object system entity	A *concept* that can be described, observed and measured and is relevant for the *problem* at hand. They are also characterized by *relevant invariant quantities* and/or *relevant variable quantities*.
Object system relation	A *relation* that connects all *object system entities* directly or indirectly with each other.
Wanted solution	A description of requirements to a solution of the *problem* wanted by the *problem owner* and other *actors*.
Problem scenario	One or more future *wanted solutions* in terms of an expected or wanted situation.
Problem complexity	The structure of the problem. It can be an *atomic problem* (i.e. single undividable) or a *composite problem* (dividable in *atomic problems*)
Atomic *problem*	A *problem* consisting of a single part, which is undividable in social and technical contexts and cannot be split in smaller *problems* without losing the characteristics of a *problem*.
Composite *problem*	A *problem* made up of complicated and related parts, each representing an *atomic problem*.
Problem description	Textual description of the *problem* at hand.
Functional *knowledge*	Knowledge on the function of the *object system*. Note 1. Structural function is defined in ontological layer 2.
Ecological *functional knowledge*	The *functional knowledge* that describes the ecological function of natural systems, e.g. *ecosystems*. Note 1. *Structural function* is defined in *ontological layer* 2.
Economic *functional knowledge*	The *functional knowledge* that describes the economic aspects of all or some *object system entities* in general and to the *stakeholders* specifically. Note 1. *Functional knowledge* is defined in *ontological layer 1*.
Problem subject	The class of problem topics, e.g. global heating, deforestation, algal blooms. Note 1. This should not be confused with problem domain.

Pollution	A *problem subject* including the introduction of contaminants into an environment. Note 1. *Problem subject* is defined in *ontological layer 1*. Note 2. Adapted from Wikipedia.
Disturbed ecosystem	A *problem subject* in which an ecosystem is pronouncedly changed, due to a (temporary) change in average environmental conditions. Note 1. *Problem subject* is defined in *ontological layer 1*. Note 2. Adapted from Wikipedia.
Nature conservation	A *problem subject* including the protection, preservation, management, or restoration of wildlife and of natural resources such as forests, soil, and water. Note 1. Problem subject is defined in *ontological layer 1*. Note 2. Adapted from the Free Dictionary (http://www.thefreedictionary.com/conservation).
Climate change	A *problem subject* including a change in the world's climate. Note 1. Problem subject is defined in *ontological layer 1*. Note 2. Adapted from the Free Dictionary (http://www.thefreedictionary.com/climate+change).
Problem context	The aspects and situation relevant for the *problem*.
Ownership	Aspect of the *problem context* determining who the owner of the *problem* is.
Private person ownership	Instance of *ownership* referring to a juridical entity of the type 'natural person' conform private law. Note 1. In the Netherlands 'private law' is called 'Burgerlijk Wetboek'.
Private company ownership	Instance of *ownership* referring to a juridical entity of the type 'private company', whose liability is limited by shares. Note 1. In the Netherlands private companies can be divided in several categories e.g. BV, NV, CV, etc.
Public ownership	Instance of *ownership* referring to group of *private persons* with interest in the *problem*.
National, Governmental ownership	Instance of *ownership* referring to an organization that is the governing authority of a political unit of the type country.
Province, State ownership	Instance of ownership referring to an organization that is the governing authority of a political country Part of a *country*.
Local, municipal	Instance of ownership referring to an organization that is the governing authority of a political country Part of a *province*, *state*.
Interest group ownership	Organized group of *private persons* with interest in the *problem*.
NGO ownership	*National Governmental* Organization, instance of ownership referring to an organization that is the governing authority of a political unit of the type country. Note 1. In Dutch 'Zelfstandige bestuursorgaan'.
Legal responsibility	Aspect of the *problem context* determining responsibility for a problem according to the law.
Private company responsibility	Instance of *legal responsibility* referring to a juridical entity of the type 'private company', whose liability is limited by shares. Note 1. In the Netherlands private companies can be divided in several categories e.g. BV, NV, CV, etc.
Authorities responsibility	Instance of *legal responsibility* referring to several juridical entities at national or sub national level (province, state, local, municipal).
NGO responsibility	Instance of *legal responsibility* referring to an organization that is the governing authority of a political unit of the type country.

Assessment	Aspect of the *problem context* determining the way problem solving is assessed and checked.
Peer review	Aspect of the *problem context* determining *assessment* by peers.
Authorities	Aspect of the *problem context* determining *assessment* by governmental, provincial/state or local/municipal juridical entities.
Societal assessment	Aspect of the *problem context* determining *assessment* by (interested groups of) the society.
Press assessment	Aspect of the *problem context* determining *assessment* by newspapers, radio and television journalists.
Problem solving methodology	Methodology that can be useful to *problem solving*. Note 1. Methodology is defined in the meta-ontology (Chapter 5).
Problem domain	*Domain* to which a *problem* belongs. Note 1. A *problem* can also belong to more than one *domain*.
Domain knowledge	Knowledge originating from and belonging to a specific *knowledge domain*.
Knowledge *domain*	Synonym for *domain*. Note 1. *Domain* is defined in the defined in the ontology (Chapter 5).
Structural *knowledge*	Knowledge on the *object system structure* (*object system entities* and *object system relations*) of the *object system*.
Process knowledge	1. Knowledge on the processes of the *object system*. 2. In science, a *process* is any method (or event) that results in a transformation in a physical or biological object, a substance or an organism. 3. In business *process* is a collection of related structural activities that produce something of value to the organization, its stakeholders or its customers. It is, for example, the process through which an organization realizes its services to its customers. Note 1. Source: of meaning 2 and 3: Wikipedia.
Physical knowledge	Knowledge from natural sciences, e.g. *physics*, *chemistry*, *geology* and *biology*.
Physics	The branch of science concerned with the nature and properties of matter and energy. Note 1. Source: Oxford Dictionary.
Chemistry	The branch of science concerned with the properties and interactions of the substances of which matter is composed. Note 1. Source: Oxford Dictionary.
Geology	The science, which deals with the physical structure and substance of the earth. Note 1. Source: Oxford Dictionary.
Biology	The scientific study of living organisms. Note 1. Source: Oxford Dictionary.
Management knowledge	Knowledge belonging to *management science*.
Management science	Discipline using mathematics, and other analytical methods, to help make better business decisions. Note 1. Source: Wikipedia. Note 2. Synonym: Operations Research (OR).
Economic knowledge	Knowledge belonging to *economics*.

Economics	The branch of knowledge concerned with the production, consumption, and transfer of wealth. Note 1. Source: Oxford Dictionary.
Sociologic knowledge	Knowledge belonging to *sociology*.
Sociology	The study of the development, structure, and functioning of human society. Note 1. Source: Oxford Dictionary.
Multi-domain knowledge	Knowledge from more than one *knowledge domain*.
Control space	Space consisting of decision parameter of the *object system* that can be changed in order to change the *object system* to arrive at a *wanted solution* of the *problem*. Note 1. *Decision parameter* is defined in Chapter 9 and Appendix G. Note 2. In Operations Research (OR) often called *control variable*.

F.2 Ontological layer 2: problem knowledge for quantitative models

Ontological layer 2 (specialized problem knowledge) is more specialized than *ontological layer 1*, as it is the bridge between rather generic concepts and relations of the generic *ontological layer 1* and *ontological layer 3*, which contains actual (scientific) knowledge, in the instance layer. It contains all concepts depicted in the structure diagram of Figure 8-4 of Chapter 8 and all concepts are defined in Table F-2.

Table F-2. Problem knowledge for quantitative models (*ontological layer 2*). All terms in *italic* are defined components of the meta-ontology (Chapter 5) and *ontological layer 1*.

Concepts	Meaning
Object system entity	A *concept* that can be described, observed and measured and is relevant for the *problem* at hand. They are also characterized by *relevant invariant quantities* and/or *relevant variable quantities*. Note 1. Is defined in *ontological layer 1*.
Static *object system entity*	*Object system entity* that does not change its value and characterizes an *object system entity.* Note 1. Synonym: 'parameter', not to be confused with *model parameter* type. Note 1. *Object system entity* is defined in *ontological layer 1*.
Variable *object system entity*	*Object system entity* able to assume different numerical values that characterizes an *object system entity*. Note 1. Synonym: 'variable', not to be confused with *model variable type*. Note 1. *Object system entity* is defined in *ontological layer 1*.
Observable *object system entity*	*Object system entity*, which value(s) can be measured/observed in the *object system.* Note 1.Example/synoniem: field data, experimental data, etc. Note 1. *Object system entity* is defined in *ontological layer 1*.
Not observable *object system entity*	*Object system entity*, which value(s) cannot be observed/measured in the *object system*. Note 1. Example: quality of life. Note 1. *Object system entity* is defined in *ontological layer 1*.
Observed *object system entity*	*Observable object system entity*, which is actually observed/measured in the *object system*. Note1. To be used to compare model results (Chapter 9) with for different purposes, including calibration (Chapter 6), validation (Chapter 6). Note 1. *Object system entity* is defined in *ontological layer 1*.

Not observed *object system entity*	*Observable object system entity*, which is not observed/measured in the *object system*. Note 1. Because it is an *Observable object system entity* it can be observed/measured but this not yet done or no data are known from the observation/measurement. Note 1. *Object system entity* is defined in *ontological layer 1*.
Object system aggregation level	General view on the level of detailing the *object system*.

F.3 *Ontological layer 3:* problem knowledge for simulation models

Ontological layer 3 (problem knowledge for simulation models) is more specialized than *ontological layer 2* (problem knowledge for quantitative models), but less specialized than *ontological layer 4*. This *ontological layer* contains all concepts depicted in the structure diagram of Figure 8-5 of Chapter 8 and all concepts are defined in Table F-3.

Table F-3. Problem knowledge for simulation models (*ontological layer 3*). All terms in *italic* are defined components of the meta-ontology (Chapter 5), *ontological layers1* and *ontological layer 2*.

Concepts	Meaning
Process knowledge in *science*	Any method (or event) that results in a transformation in a physical or biological object, a substance or an organism. Note 1. Source: Wikipedia. Note 2. *Science* is defined in the meta-ontology of Chapter 5.
Physics *process*	The science of matter and its motion, as well as space and time. It uses concepts such as energy, force, mass, and charge. Physics is an experimental science, creating theories that are tested against observations. Broadly, it is the general scientific analysis of nature, with a goal of understanding how the universe behaves. Note 1. Source: Wikipedia. Note 2. 'Physical object' should here be seen as part of the *object system*, or more specifically as an *object system entity*.
Transport *process*	The movement of people, goods and matter from one place to another. The term is derived from the Latin *trans* ("across") and *portare* ("to carry"). Note 1. Source: Wikipedia.
Water quantity transport	The movement of water quantities by means of a *transport process*.
Dissolved matter transport	Transport of dissolved matter by means of *water quantity transport*.
Particulate matter transport	Transport of particulate matter by means of *water quantity transport*.
Chemical *process*	Any process determined by the atomic and molecular composition and structure of the substances involved. Note 1. Synonym: chemical change, chemical action. Note 2. Source: Wikipedia.
Endothermic reaction	A chemical reaction in which the products have more energy than the reactants, and thus a net input of energy, usually in the form of heat, is required. Note 1. Source: Wikipedia.
Exothermic reaction	A chemical reaction that releases heat and therefore the opposite of an *endothermic reaction*. Note 1. Source: Wikipedia.
Biological *process*	Any process in which living organisms are involved.

Ecological *process*	Any process concerning the distribution and abundance of living organisms and how these properties are affected by interactions between the organisms and their environment. Note 1. Source: Wikipedia.
Physiological *process*	Any process concerning the mechanical, physical, and biochemical functions of living organisms.
Biochemical *process*	
Lumped *process*	A combination of *instances* of *process knowledge*.
What is processed	The subject of some process.
Substance	*Instance* of *what is processed*, being 1. That what has mass and occupies space: matter 2. A material of a particular kind or constitution. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition.
Energy	*Instance* of *what is processed* being power derived from physical or chemical resources to provide light and heat or to work machines. Note 1. Source: Oxford Dictionary.
Money	*Instance* of *what is processed*, being medium of exchange in the form of coins and banknotes. Note 1. Source: Oxford Dictionary.
Knowledge	*Instance* of *what is processed*, being information and skills acquired through experience or education. Note 1. Source: Oxford Dictionary.

F.4 Ontological layer 4: problem knowledge for application domains

Ontological layer 4 (problem knowledge for application domains, here instantiated for modelling bivalve ecophysiology) is more specialized than *ontological layer 3* (problem knowledge for simulation models), as it contains scientific, very detailed factual knowledge. This *ontological layer* contains all concepts depicted in the structure diagram of Figure 8-6 of Chapter 8 and all concepts are defined in Table F-4.

Table F-4. Problem knowledge for application domains, here instantiated for modelling bivalve ecophysiology (*ontological layer 4*). All terms in *italic* are defined components of the meta-ontology (Chapter 5), *ontological layer 1*, *ontological layer 2* and *ontological layer 3*. Concepts between square brackets ('[' and ']') are *units* of that concept.

Concepts	Meaning
Ecosystem/organization	Instance of an *object system aggregation level*, being a biological community of interacting organisms and their physical environment. An *organization* is an organized body of people with a particular purpose, e.g. a business or a social community. Note 1. Source: Oxford Dictionary. Note 2. Object system aggregation level is defined in *ontological layer* 2.
Population	Instance of an *object system aggregation level*, being a community of interbreeding organisms. Note 1. Source: Oxford Dictionary. Note 2. Including humans. Note 3. Object system aggregation level is defined in *ontological layer* 2.
Organism	Instance of an *object system aggregation level*, being 1. An individual animal, plant, or single-celled life form. 2. A whole with interdependent parts. Note 1. Source: Oxford Dictionary. Note 2. Object system aggregation level is defined in *ontological layer* 2.

Organ	Instance of an *object system aggregation level*, being a distinct part of an animal or plant adapted for a particular function, for example the heart or kidneys. Note 1. Source: Oxford Dictionary. Note 2. Object system aggregation level is defined in *ontological layer* 2.
Aggregated entity	Instance of an *object system aggregation level*, being *object system entities* combined to simplify the view on the *object system*.
Functional compartment	Instance of an *object system aggregation level*, being a simplification of many *organs* into a few components. Organs are combined in a functional compartment because of their common functional role. Note 1. Synonym: ‘Functional part” Note 2. ‘Functional group’ is another synonym, but only used for a group of *organisms*.
Somatic compartment	Total body of bivalve, except storage compartment, reproductive compartment and shell compartment.
Storage compartment	All *glycogen* in a bivalve.
Reproductive compartment	Reproductive *organs* (*gonads*) and *reproductive cells* (*gametes*)
Shell compartment	Consists of two parts the *inorganic part* (i.e. the shell) and an *organic part* that produces the inorganic part
Glycogen	A polysaccharide $(C_6H_{10}O_5)_n$ carbohydrate that is the main form of carbohydrate in animals. It can easily be transferred to glucose.
Habitat	The place where a particular species lives and grows. It is essentially the environment—at least the physical environment—that surrounds (influences and is utilized by) a species population. Note 1. Source: Wikipedia. Note 2. From the Latin for "it inhabits".
Epibenthical	With byssus threads connected to each other or some substrate.
Infaunal	Buried in upper sediment.
Seston	All particles in water. Note 1. Synonyms: Total Particular matter, TPM, seston.
[*Seston*]	[$g.m^{-3}$]
Total Particulate matter	Seston. Note 1. Synonyms: TPM, seston.
TPM	*Total Particulate Matter*.
[*TPM*]	[$g.m^{-3}$]
Suspended Particulate Matter	Seston Note 1. Synonyms: TPM, seston. Note 2. Emphasis is here on the fact that there is a balance between TPM in the water column and suspended seston on the bottom. Wind and waves have a significant influence of the distribution of seston at different depths in the water column.
Silt	Inorganic part of seston.
[*Silt*]	[$g.m^{-3}$]
Particulate Organic Matter	Organic Part of seston. Note 1. POM.
POM	*Particulate Organic Matter*
[*POM*]	[$gDW.m^{-3}$]
Particulate Organic Carbon	Organic carbon in *seston*.
POC	*Particulate Organic Carbon*.
[*POC*]	[$gC.m^{-3}$]

Plankton	Free floating organisms, which movements are mainly due to currents, wind, etc. Their size, ranges from <0.2 [μm] to 20,000 [μm]. Plankton consists of *phytoplankton*, i.e. small plants, *zooplankton*, i.e. small animals and bacterioplankton, i.e. bacteria.
Algae	Any of various chiefly aquatic, eukaryotic, photosynthetic organisms, ranging in size from single celled forms to giant kelp. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition. Note 2. Algae were once considered to be plants but are now classified separately, because they lack true roots, stems, leaves and embryos.
Phytoplankton	Single celled photosynthetic *plankton*. Note 1. Synonym: *Algae* and *Algal cells*.
Algal cells	*Phytoplankton*
Chlorophyll	*Chlorophyll* concentration is used as a measure for *phytoplankton* concentration as it is available in living phytoplankton.
[*Chlorophyll*]	A green photosynthetic pigment found in plants, algae, and cyanobacteria. Note 1. Chlorophyll is used to measure the phytoplankton biomass.
Diatom	*Phytoplankton* with silicate skeleton.
Flagellate	*Phytoplankton* without silicate skeleton.
Phaeocystis (spss.)	*Flagellate phytoplankton* species forming *colonies* that have no nutritional value for bivalves, as the colonies are too big.
[*Phaeocystis* concentration]	[*cell.l*$^{-1}$]
Colony	Framework of tightly coupled *Phaeocystis (spss.)* cells.
Number of cells in a *colony*	Number of *Phaeocystis (spss.)* cells in a colony.
[*Colony size*]	[*Number of Phaeocystis cells*.colony^{-1}]
Total *colony* surface	The outer surface of the *Phaeocystis (spss.) colony*.
[*Total colony surface*]	[*mm*^{2}]
Zooplankton	Animal part of *plankton*.
FytoPOC	*Phytoplankton* part of *POC*.
FytoPOM	*Phytoplankton* part of *POM*.
Detritus	Dead *Particulate Organic Material*. Note 1. Without living *algae*. Note 2. The quality of *detritus* as *food* source for other organisms depends mainly of its age, as it consists mainly of dead *algae* and the proteins (containing large amounts of nitrogen) are easier to decay than carbohydrates. Old *detritus* consists therefore of poor *particulate* and *dissolved organic matter* with lower nutritional values.
Refractory *detritus*	*Detritus* that is hard to *decompose*. Note 1. Typically older *detritus*, of which the *labile detritus* part is already *decomposed*.
Labile *detritus*	*Detritus* that easily *decomposes*. Note 1. Typically younger *detritus*, containing large amounts of recently died *algae*.
Food	*Phytoplankton* (including small parts of *zooplankton*) and *detritus*. Note 1. Food refers here only to *food* for *bivalves*.
Food quality	The more living or recently died *phytoplankton* in *food*, the higher the quality.

Mollusk	Any of numerous chiefly marine invertebrates of the phylum Mollusca, typically having a soft unsegmented body, a mantle, and a protective calcareous shell and including the edible shellfish and the snails. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition.
Bivalve	A *mollusk*, such as an oyster or a clam, that has a *shell* consisting of two hinged valves.
Blue mussel	*Mytilus edulis* L.
Mytilus edulis L.	A marine bivalve mollusk of the family Mytilidae, a blue-black species raised commercially in Europe. Mussels are often found attached to rocky surfaces or the side of ships. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition (adapted). Note 2. There are also other mussels, mainly marine but also some freshwater species.
Cockle	*Cerastoderma edule* (L.)
Cerastoderma edule (L.)	A *bivalve mollusk* of the family Cardiidae, having rounded or heart-shaped *shells* with radiating ribs.
Biomass	The total mass of living matter within a given unit of environmental area. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition.
[*Biomass*]	[gDW]
Dry Weight	Quantity determining the *biomass*, after a treatment in a stove of 60, 80 or 105 ºC until no weight loss can be observed (12-24 hours or even more), removing all water content. Note 1. Synonyms: DW. Note 2. The *carbon content* (C) or the *nitrogen content* (N) can also be used as a measure for biomass. Typically the carbon content = 0.5 DW.
Fresh weight	*Quantity* determining the *biomass* after removing clinging water.
Body size	Size of an individual, for *bivalves* commonly expressed as *biomass*, *dry weight* or *fresh weight*.
[Carbon content]	[gC]
[Nitrogen content]	[gN]
Total Wet Weight	Total weight of shell including (wet) *biomass* of the *bivalve* organism.
Cooked Flesh Weight	(Wet) weight after cooking the *bivalve* organism.
Shell	*Bivalves* have two matching calciferous halve *shells* joined together at a hinge and held closed by a set of muscles. Like other *mollusks*, *bivalves* possess a hard exterior *shell* and no internal skeleton.
Organic part of *shell*	A small fraction of the *shell* consisting organic material.
Inorganic part of *shell*	The large, inorganic part of the *shell*.
Organic *Shell* Weight	Weight of *organic part of shell*.
Dry *Shell* Weight	Weight of the organic and inorganic part of the shell, after removal of water. Note 1. See Ash Free Dry Weight.
Shell length	Length of the *shell*.
[*Shell length*]	[mm]
Digestive tract	Organs to pass the food through an organism, including *mouth*, *esophagus*, *stomach*, *intestine(s)* and *anus*.
Gut	*Stomach* and *intestines*
Gut content	The volume of the *gut*.
Gut passage time	The time *food* needs to pass the *gut*.

[*Gut passage time*]	[hours]
Esophagus	*Organ*, by which food is transported from mouth to stomach. Part of the digestive tract. Note 1. Synonym: Esophagus.
Stomach	*Organ*, in which food is stored and digested. Part of the digestive tract.
Intestine(s)	*Organ* (between stomach and anus), in which the digestion and absorption of digested food occurs. Part of the digestive tract.
Gill	*Organ* to filter *food* particles from the passing water and to exchange oxygen (from water to *bivalve* blood) and carbon dioxide (from blood to water).
Anus	The end of the *intestine(s)*, where *faeces* leave the body.
Byssus threads	The long fine silky filaments excreted by several mollusks by which they attach themselves. They range to 6 cm in length. Note 1. Source: Wikipedia. Note 2. *Byssus threads* are mainly produced as a response to high currents, and after replacement of the animals. In the knowledge perception in *ontological layer* 3, byssus thread production is included in the organic shell growth.
Metabolism	1. The biochemical modification of chemical compounds in living *organisms* and cells. This includes biosynthesis of complex organic molecules (anabolism) and their breakdown (catabolism). Metabolism usually consists of sequences of enzymatic steps, also called metabolic pathways. The total metabolism comprises all biochemical processes of an organism. The cell metabolism includes all chemical processes in a cell. 2. The chemical *processes* occurring within a living *cell* or *organism* that are necessary for the *maintenance* of life. In *metabolism* some substances are broken down to yield energy for vital *processes* while other substances, necessary for life are synthesized. Note 1. Source meaning 1: Wikipedia. Note 2. Source meaning 2: The American Heritage Dictionary of the English Language, Fourth Edition (adapted).
Filtration	The process in which *seston* are retained by the gills.
Filtration rate	The rate at which *seston* are retained by the gills.
[filtration rate]	[$g.d^{-1}$]
Clearance	The process in which pumping water through the *gills*.
Clearance rate	The volume of water that's is cleared from *seston* per unit of time.
[Clearance rate]	[$l.h^{-1}.organism^{-1}$]
Clearance depression	Reduction of the *clearance rate* due to one of the following causes: • Too high *seston* concentrations (too a high extent consisting of (inorganic) silt. • Too much *Phaeocystis colonies* in *seston*. • Too high *food quality*, i.e. mainly living *algal cells* (*phytoplankton*). Note 1. Causes 1. and 2. occur in natural conditions, depending of the ecosystem; cause 3. is only observed in experiments.
Pre-ingestive food selection	The process of enriching food by bivalves in case of low food quality. Note 1. Synonyms: food selection and pre-ingestive selection
Food selection	*Pre-ingestive food selection*
Pre-ingestive selection	*Pre-ingestive food selection*
Selection efficiency	The efficiency of *pre-ingestive food selection*, expressed as: 1-(organic fraction in pseudo-faeces)/(organic fraction in seston)
[*Selection efficiency*]	[-]

Pseudo-faeces	The ejected part of the retained *seston* particles during *filtration*.
Pseudo-*faeces* production	The production of *pseudo-faeces*. Note 1. Sometimes referred to as 'rejection' and 'rejection rate'.
[Pseudo-*faeces* production]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Mucus	Slimy excretion used to pack non-*food* particles as *pseudo-faces*.
Mucus production	Production of *mucus*.
Mucus production rate	*Mucus production* per unit of time.
[*Mucus production rate*]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Ingestion, ingest	The process in which retained *seston* from *filtration* goes from the *gills* into the *gut*.
Ingestion rate	What is *ingested* per time unit.
[*Ingestion rate*]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Absorption, absorb	1. The process of absorbing nutrients into the body after digestion. 2. *Ingested seston* is digested and passes the wall of the *gut* and can be used for *maintenance*, *growth* and *reproduction*. Note 1. Source of meaning 1: WordNet 2.0.
Absorption rate	What is *absorbed* per unit of time.
[*Absorption rate*]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Absorption efficiency	The efficiency of *absorption*, expressed as: 1-(organic fraction in *faeces*)/(organic fraction in ingested food)
[*Absorption efficiency*]	[-]
Faeces	Waste product from an animal's digestive tract expelled through the anus during defecation. Note 1. Source Wikipedia Note 2. Synonym: Feces.
Faeces production	Production of *faeces*.
Faeces production rate	*Faeces production* per unit of time.
[*Faeces production rate*]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Excretion, excrete	Removal of ammonium (NH4) produced as metabolic waste material. Note 1. *Excretion* is not observed in intertidal cockles. Note 2. In any physiological balance (model) a nitrogen budget should only be included if *excretion* has been observed.
Excretion rate	*Excretion* per unit of time.
[Excretion rate]	[gDW.d^{-1}.$organism^{-1}$] or [gC.d^{-1}.$organism^{-1}$]
Growth, grow	Increase of the organic and inorganic part of an organism.
Growth rate	Growth per unit of time. Note 1. Growth typically refers to growth of the organic part of an organism. Note 2. See also shell growth.
[Growth rate]	*Growth* per unit of time.
Scope for *growth*	Difference between *absorption* and *respiration* expressed as fraction of body size.
Shell growth	*Growth* of the *shell*, usually expressed as *growth* in *shell length*, consisting of: • *Growth organic part of shell* • *Growth inorganic part of shell*
Growth organic part of shell	*Growth* of the organic part of the shell, including *byssus threads*.
Growth inorganic part of shell	The *organic part of the shell* is responsible for growth of the *inorganic part of the shell*.
Maintenance	The physiological *processes respiration* and *excretion*, necessary to stay alive.
Respiration, respire	The *process* in which oxygen is used to oxidize carbohydrates to produce energy for *maintenance* and *growth*.

Respiration rate	*Respiration* per unit of time.
Rest *respiration*	Respiration when the bivalve is not actively pumping.
Active *respiration*	In addition of the rest *respiration*, bivalves will use more energy when active. This *respiration* is coupled to activity (e.g. *ingestion*, *filtration* and *clearance*).
[*Respiration rate*]	[mg $O_2.h^{-1}$. organism^{-1}] or [gC.d^{-1}. organism^{-1}]
Reproduction	The sexual or asexual *process* by which *organisms* generate new individuals of the same kind. Note 1. Source: The American Heritage Dictionary of the English Language, Fourth Edition.
Gonad	Reproductive *organ*. Note 1. Most bivalves have two sexes. Some change their sex during their lifetime, starting as males and ending their lives as females.
Gonad weight	Mass of reproductive organs including weight of reproductive cells. Note1. Sometimes gonad weight is expressed as fraction or percentage of bodyweight.
Reproductive cell	Eggs and sperm, together called gametes
Gamete	Reproductive cell
Gametogenesis	Forming of *reproductive cells*
Spawn, spawning	Release of *gametes*. Note 1. Synonym: Spat.
Spawning rate	*Spawning* per unit of time.
[*Spawning rate*]	[gC.d^{-1}.organism^{-1}] Note 1. In case of mussels, in which nitrogen metabolism is relevant, also [gN.d^{-1}. organism^{-1}]
Time of *spawning*	Moment at which spawning occurs.
[*Time of spawning*]	[Day number]
Decomposition, decompose	Decay of organic material by bacteria or other living organisms.

F.5 *Ontological layer 5:* problem knowledge for projects

F.5.1 Template

The knowledge of *ontological layer 4* has been used in several scientific projects, all using a simulation model to understand a biological system, related to bivalve ecophysiology. The structural concepts presented in *ontological layer 5* are rather generic as they aim to provide vocabulary to describe a series of science related projects on bivalve ecophysiology and ecological process. Table F-5 gives an overview of the structural concepts in *ontological layer 5*.

Table F-5. Problem knowledge ontology for specific application domain projects (*ontological layer 5*). All terms in *italic* are defined components of other *ontological layers* (0-4).

Concepts	Meaning
Project description	Name and period of research project(s).
Object system name	Name of the *object system*.
Object system description	Description of the main characteristics of the *object system*.
Object system aggregation level	General view on the level of detailing the *object system*. Note1. Defined in *ontological layer 2*.
Project purpose	Purpose of the *project*.
Project context	Context of the *project* in terms of budget, sponsor and reference(s).
Project references	References to journal papers, reports and other explicit scientific publications.
Knowledge *specializations*	Adaptations and/or specializations of the knowledge content in a less specialized *ontological layer*.

F.5.2 Instantiations for bivalve ecophysiology projects

F.5.2.1 Simplified ecophysiology of mussels and cockles in an ecosystem

In the first case study mussels and cockles are entities in an ecosystem (*object system*) with a substantial effect of that ecosystem. A part of the ecophysiological knowledge of bivalves as described in section F.3 has been gathered in a large ecosystem study in the Oosterschelde SW Netherlands (Klepper, 1989, Herman and Scholten, 1990, Scholten *et al.,* 1990, Van der Tol and Scholten, 1992, Klepper *et al.,* 1994, Scholten and Van der Tol, 1994, Van der Tol and Scholten, 1998). The project characteristics are described in Table F-6, as instance of the concepts in Table F-5.

Table F-6. Problem knowledge ontology instantiation for specific application domain projects (*ontological layer* 5). This table is an instantiation of Table F-5, filled for a specific problem, here BALANS/EOS. All terms in italic are defined concepts of the meta-ontology (Chapter 5) or of *ontological layers 0*, *1*, *2*, *3*, *4*).

Concepts	In this project
Project description	Ecosystem study consisting of 2 projects: • BALANS (1980-1988) and • EOS (1988-1991)
Object system name	Oosterschelde ecosystem
Object system description	• The Oosterschelde estuary (or coastal bay) is located in SW Netherlands. After completing the storm surge barrier and associated coastal works, its area is 351 km^2 (averaged over the tide; before the storm surge barrier it was 452 km^2) with an average depth of 7.8 m and average volume of 2740 million m^3. • Primary production *phytoplankton* and *phytobenthos* in the Oosterschelde is of the same order of magnitude as *plankton* and *detritus* transported by tidal dislocation (Scholten *et al*., 1990). *Phytoplankton*, *detritus* and *suspended phytobenthos* are *food* for *zooplankton* and especially *cockles* and *mussels*. The latter two species control *nutrient* and light limited *phytoplankton* production (Herman and Scholten, 1990). • In these projects the *ecophysiology* of *bivalves* has been perceived in a simplified form, mainly *respiration* and *excretion*, as significant processes in the carbon and nitrogen *ecosystem* budget.
Object system aggregation level	Ecosystem, but with some detail at organism level.
Project purpose	Ecosystem study to investigate the effects of the Oosterschelde storm surge barrier, resulting in ecosystem model SMOES (see Chapter 9).
Project context	The BALANS project (M€ 5) and the EOS project (M€ 5) were organized by RIKZ (formerly known as DGW) and extensively described by Nienhuis and Smaal (1994).
Project references	• Klepper and Scholten (1988) • Klepper (1989) • Klepper *et al*. (1991) • Nienhuis and Smaal (1994) • Klepper *et al*. (1994) • Scholten and Van der Tol (1994)
Knowledge specializations	All mussels are assumed to occur in subtidal habitats and all cockles on intertidal flats, which are above water level at low tide.

F.5.2.2 The ecophysiology of Mytilus edulis *L.*

In the next three case studies the same knowledge on ecophysiology of mussels is tested in object systems differing completely in ecological conditions.

The first of these three case studies aimed at simulating the physiological response of mussels (*Mytilus* edulis L.) to the natural ecological conditions, found in the Oosterschelde ecosystem, SW Netherlands, using the ecophysiological knowledge of section F.3. The knowledge of section F.3 has been used to develop a simulation model[1] for the growth of a single mussel based on the ecophysiological response of mussels to ecological conditions (Smaal, 1997, Smaal and Scholten, 1997). The project characteristics are described in Table F-7, as instance of the concepts in Table F-5.

Table F-7. Problem knowledge ontology instantiation for specific application domain projects (*ontological layer* 5). This table is an instantiation of Table F-5, filled for a specific problem, here EMMY. All terms in italic are defined concepts of the meta-ontology (Chapter 5) or of *ontological layers 0*, *1*, *2*, *3*, *4*).

Concepts	In this project
Project description	EMMY
Object system name	Single mussel (Mytilus edulis *L.*) in Oosterschelde estuary, SW Netherlands, 1982-1983.
Object system description	• The *ecophysiological* knowledge on a single *mussel*, *Mytilus edulis L.* includes feedback mechanisms in the acquisition and *metabolism* of natural *food* sources, and partitioning of carbon and nitrogen to the somatic compartment, storage, organic shell matrix, blood, and gametes before and after spawning. • The knowledge must enable to develop a model capable to describe the *ecophysiological* response of a single subtidal mussel to ecological conditions in the Oosterschelde estuary.
Object system aggregation level	Organism level with details on organ / tissue level.
Project purpose	Developing a model for Smaal (1997), in order to: • Understand mussel ecophysiology • Combine experimental and field knowledge • Identify knowledge gaps
Project context	SEO[2] funding by RIVO-DLO.
Project references	• Smaal (1997) • Smaal and Scholten (1997)
Knowledge specializations	1. The mussel is assumed to occur in a subtidal habitat. 2. *Mussel* blood is assumed to play a role in *allocation* of *absorbed food*.

The second of these three case studies aimed at a more formal project to present the ecophysiological knowledge on mussels in a severely peer reviewed journal (Scholten and Smaal, 1998). The mussels had to cope in this study to the natural ecological conditions, found in three ecosystems, differing mainly in the amount of food available for the mussels. The result was an updated version of the simulation model[3] in the first of the three case studies on mussel ecophysiology. The project characteristics are described in Table F-8 as instance of the concepts in Table F-5.

Table F-8. Problem knowledge ontology instantiation for specific application domain projects (*ontological layer* 5). This table is an instantiation of Table F-5, filled for a specific problem, here JEMBE. All terms in italic are defined concepts of the meta-ontology (Chapter 5) or of *ontological layers 0*, *1*, *2*, *3*, *4*).

Concepts	In this project
Project description	JEMBE
Object system name	Single mussel (Mytilus edulis *L.*) in 3 ecosystems: • The western part of the Oosterschelde, SW Netherlands, 1982-1987; • The bay of Marennes-Oléron, France, 1983-1984; • Upper South Cove, Canada, 1991.
Object system description	• The *ecophysiological* knowledge on a single *mussel*, *Mytilus*

[1] The simulation model is EMMY, version 1.6, which is described in Chapter 9 and in Appendix G.
[2] SEO means ‘Strategische Expertise Ontwikkeling’ (strategic expertise development).
[3] The simulation model is EMMY, version 2.0, which is described in Chapter 9 and in Appendix G.

	edulis L. includes feedback mechanisms in the acquisition and *metabolism* of natural *food* sources, and partitioning of carbon and nitrogen to the somatic compartment, storage, organic shell matrix, blood, and gametes before and after spawning.
	• The knowledge must foster a model capable to describe the *ecophysiological* response of a single subtidal mussel to ecological conditions in
	  ○ The Oosterschelde estuary (SW Netherlands)
	  ○ The bay of Marennes-Oléron (France)
	  ○ Upper South Cove (Canada)

Object system aggregation level	Organism level with details on organ / tissue level.
Project purpose	Get a peer reviewed paper on the knowledge and associated model in a SCI-journal (Journal of Experimental Marine Biology and Ecology), which: Reflects mussel ecophysiology Combine experimental and field knowledge Identify knowledge gaps Is sufficient to be used under substantial different ecological conditions
Project context	Partially funded by EU Concerted Action AIR3-CT94-2219 'Trophic capacity of coastal zones for rearing oysters, mussels and cockles'. Also partially SEO funded by RIVO-DLO.
Project references	• Scholten and Smaal (1998)
Knowledge specializations	1. The mussel is assumed to occur in a subtidal habitat. 2. *Mussel* blood is assumed to play a role in *allocation* of *absorbed food*.

In the third of these three case studies the lessons learned from the two previous studies has been tested in (semi) controlled mesocosms experiments, leading to a final adaptation of the expertise on bivalve ecophysiology as described in section F.3 and of the resulting simulation model[4]. This third project on the ecophysiology of mussels focused on extending the scope of the EMMY model to experimental conditions in mesocosm[5] experiments (Scholten and Smaal, 1999). The project characteristics are described in Table F-9 as instance of the concepts in Table F-5.

Table F-9. Problem knowledge ontology instantiation for specific application domain projects (*ontological layer* 5). This table is an instantiation of Table F-5, filled for a specific problem, here Mesocosm. All terms in italic are defined concepts of the meta-ontology (Chapter 5) or of *ontological layers 0*, *1*, *2*, *3*, *4*).

Concepts	In this project
Project description	Mesocosm
Object system name	Single mussel (Mytilus edulis *L.*) in 6 mesocosms experiments with 3 levels of nutrients.
Object system description	A series of nutrient loading experiments was carried out for a period of 7 month with 6 land-based mesocosms, situated at the field station of the National Institute for Coastal and Marine Management (RIKZ) near the mouth of the Oosterschelde estuary (SW Netherlands). The mesocosms consisted of black solid polyethylene tanks (height 3 m, width 1.2 m, and volume 3 m3). Water in the mesocosms was continuously mixed with a rotating mixer. Each tank has a 150 l sediment container at the bottom, but the benthic grazers are held in separate benthos chambers (16 l each) in order to measure the flows through these compartments more easily. In the tanks the light climate

[4] The simulation model is EMMY version 2.7, which is described in Chapter 9 and in Appendix G.
[5] http://www.seagrant.sunysb.edu/BTRI/btriterms.htm#m: a 'mesocosms' is an experimental apparatus or enclosure designed to approximate natural conditions, and in which environmental factors can be manipulated. A description of the mesocosms used for this project can be found at http://www.nioo.knaw.nl/cemo/phase/mesocosm/mesophase.htm.

	and turbulence were similar to natural conditions in the Dutch coastal zone of the North Sea. The water had a residence time of 30 days during the 7-month experiments and was pumped through benthos grazing chambers at a rate of 70 l h^{-1}. Many variables were measured at regular intervals in the water column and at the entrance and outlet of the benthos chambers. A full description of this experiment is given in Prins *et al*. (1998). Three levels of nutrient loading were used to simulate different scenarios for the reduction of anthropogenic nutrient load to the coastal zone. The three levels of nutrient concentrations were defined in proportion to the levels in the coastal zone in the period 1980-1987 as this was considered as a period of high eutrophication. In 6 mesocosms 3 eutrophication reduction scenarios were studied in replicates. In two mesocosms the nutrient concentrations were relatively high (H), two had medium (M) concentrations and two had low (L) concentrations. The H-mesocosms contained 90 % of observed North Sea nitrogen levels and 50 % of observed phosphorus. In the M-mesocosms both nitrogen and phosphorus concentrations were 50 % of the observed levels. The L-mesocosms received water with 25 % of the North Sea nitrogen and phosphorus concentrations. Each tank was coupled to a benthos chamber with (initially) 40 mussels. After almost 2 months 20 mussels in each tank were harvested and analyzed and the remaining mussels were analyzed at the end of the experiment after nearly 7 months. From the large body of observations only a few were utilized to provide input for the model: chlorophyll concentrations and temperatures (both measured daily), and additionally the concentrations of particulate organic carbon, particulate organic nitrogen, and total particulate matter (measured every 3 to 4 days). Measured mussel dry weight and shell length (at t = 0, at t = 2 months, and at t = 7 months) were used to evaluate the competence of the model to make accurate predictions.
Object system aggregation level	Organism level with details on organ / tissue level.
Project purpose	Extend the scope of the EMMY model in order to deal with the experimental conditions in mesocosms, including food of extremely good *food quality*, i.e. almost pure *phytoplankton*.
Project context	SEO funding by RIVO-DLO.
Project references	• Scholten and Smaal (1999) Note 1. This paper has been awarded with the Dresscher prize 1998-1999.
Knowledge specializations	The mussel is assumed to occur in a subtidal habitat. • *Mussel* blood is skipped from the list of relevant organs and tissues because it is not required for *allocation* of *absorbed food*.

F.5.2.3 The ecophysiology of Cerastoderma edule *(L.)*

Based on the experiences of the mussel projects (F.5.2.2) a similar project has been executed for cockles, *Cerastoderma edule* (L.). Cockles differ from mussels in terms of the bivalve KB (section F.3) in their habitat (at least in the Oosterschelde conditions). Mussels are always submersed, allowing them pump water along their gills continuously, while cockles live mainly infaunally, buried in the upper layer of intertidal flats, which fall dry during longer or shorter periods at low tide, depending on flat elevation. On flats typical for the present research study, cockles are submersed for 16.8 $h.d^{-1}$. This project also resulted in a simulation model[6]. The project characteristics are described in Table F-10 as instance of the concepts in Table F-5.

Table F-10. Problem knowledge ontology instantiation for specific application domain projects (*ontological layer* 5). This table is an instantiation of Table F-5, filled for a specific problem, here COCO. All terms in italic are defined concepts of the meta-ontology (Chapter 5) or of *ontological layers 0*, *1*, *2*, *3*, *4*).

Concepts	In this project
Project description	COCO
Object system name	A single *cockle* on intertidal flats in the Oosterschelde.
Object system description	The COCO-project should support the development of an ecophysiological model of the bivalve *Cerastoderma edule* that simulates individual growth and reproduction under ambient conditions in temperature and food availability in the Oosterschelde estuary, SW Netherlands. The model is a tool for the integration of ecophysiological knowledge of this species and also for carrying-capacity studies of shellfish culture and for environmental management of populations in estuarine and coastal areas.
Object system aggregation level	Organism level with details on organ / tissue level.
Project purpose	Developing a model for Rueda *et al.*, 2005, in order to Understand cockle ecophysiology Combine experimental and field knowledge • Identify knowledge gaps
Project context	This project has been supported by a Marie-Curie training research grant of the European Commission, within the project SIMCERE (FAIR GT97-4525). It has been also supported with SEO funding by RIVO-DLO.
Project references	Rueda, Smaal and Scholten (2005).
Knowledge specializations	• Opposite to mussels, cockles occur on tidal flats and are submersed for only a fraction of the day, for the Oosterschelde flats in this project this fraction is typically 0.7 (or 16.8 $h.d^{-1}$). When not submerged they stop pumping, which means that there is no food intake. Furthermore, respiration is reduced. • Nitrogen related processes are assumed to be irrelevant to cockles.

F.6 References

Herman, P.M.J., Scholten, H., 1990. Can suspension-feeders stabilise estuarine ecosystems? In: Barnes, M., Gibson, R.N. (Eds.), Trophic relations in the marine environment. Aberdeen University Press, Aberdeen, pp. 104-116.

Klepper, O. and H. Scholten, 1988. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (SW Netherlands), DGW/DIHO, Middelburg, 278 pp.

Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary, DGW-Ministry of Transport and Public Works. Wageningen University, Wageningen, 270 pp.

Klepper, O., H. Scholten and J.P.G. Van de Kamer, 1991. Prediction uncertainty in an ecological model of the Oosterschelde estuary, S.W. Netherlands. Journal of Forecasting 10, 191-209.

Klepper, O., Van der Tol, M.W.M., Scholten, H., Herman, P.M.J., 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.

Nienhuis, P.H. and A.C. Smaal (Eds.), 1994. The Oosterschelde estuary. Developments in Hydrobiology 97, H.J. Dumont, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-2817-5, Reprinted from Hydrobiologia, vols. 282/283, 597 pp.

[6] The simulation model is COCO, version 1.1, which is described in Chapter 9 and in Appendix G.

Prins, T.C., V. Escaravage, L.P.M.J. Wetsteyn, J.C.H. Peeters and A.C. Smaal, 1998. Effects of different N- and P-loading on primary and secondary production in an experimental marine ecosystem. Aquatic Ecology 33, 65-81.

Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (Cerastoderma edule L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.

Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.

Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.

Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.

Scholten, H., O. Klepper, P.H. Nienhuis and M. Knoester, 1990. Oosterschelde estuary (S.W. Netherlands): a self-sustaining ecosystem? In: D.S. McLusky, V.N. De Jonge (Eds.), North Sea-estuaries interactions. Kluwer Academics Publishers, Dordrecht, pp. 201-215.

Smaal, A.C., 1997. Food supply and demand of bivalve suspension feeders in a tidal system, Wiskunde en Natuurwetenschappen. Rijksuniversiteit Groningen, Groningen, 237 pp.

Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.

Van der Tol, M.W.M. and H. Scholten, 1992. Response of the Eastern Scheldt ecosystem to a changing environment: functional or adaptive? Neth. J. Sea Res. 30, 175-190.

Van der Tol, M.W.M. and H. Scholten, 1998. A model analysis on the effect of decreasing nutrient loads on the biomass of benthic suspension feeders in the Oosterschelde (SW Netherlands). Journal of Aquatic Ecology 31, 395-408.

Appendix G Concepts of the model ontology

Content of Appendix G, belonging to chapter 9

G.1 Ontological layer 1: generic model knowledge

In Chapter 9 a model ontology has been proposed, which is structured in so called *ontological layers* aiming to facilitate reuse of parts of the ontology. *Ontological layer 1* contains a *generic model* ontology and will be filled with *concepts* and the *relations* connecting the concepts, which are common for all *mathematical modelling paradigms*[1]. The terminology, i.e. *concepts* and *relations* in this layer should be instrumental for discussing the various instances of ontological layers 2. Each of the latter will contain terminology, i.e. *concepts* and *relations* for a specific *mathematical modelling paradigm*, e.g. simulation, optimization etc. In this section all concepts used in that ontology are defined in Table G-1. This *ontological layer* contains all concepts depicted in the structure diagram of Figure 9-3 of Chapter 9.

Table G-1. Concepts of *ontological layer 2* (generic model knowledge). All terms in *italic* are defined components of the *meta-ontology* (Chapter 5).

Concepts	Meaning
Model	To an observer B, an object A^* is a model of an object A to the extent that B can use A^* to answer questions that interest him about A. Note 1. From Minsky (1965). Note 2. Model is already defined in the *meta-ontology* of Chapter 5. Note 3. See also *first draft of a mental process model*, given in Chapter 2, Table 2-1.
Mathematical model	An equation or a set of equations representing real-world phenomena. Note 1. Source: www.mathwords.com/a_to_z.htm Note 2. Models also represent patterns found in graphs and data (in this case often called statistical models). Note 3. Usually models are not exact matches the objects or behavior they represent. A good model should capture the essential character of whatever is being modeled. Note 4. A *mathematical model* describes the *conceptual model* in a mathematical format. Note 5. *Conceptual model* is defined in *ontological layer 2* (Table G-2).
Physical model	A physical model is a smaller or larger physical copy of an object. The object being modeled may be small (for example, an atom) or large (for example, the Solar System). Note 1. Source: Wikipedia, http://en.wikipedia.org/wiki/Model_(physical). .
Representation power	That part of an *object system* that a model should be able to represent: representing an *object system structure* and/or *object system behavior*. Note 1. *Object system*, *object system structure* and *object system behavior* are defined in Chapter 8 and Appendix F.
Behavioral *representation power*	The part of an object system behavior that is represented by a model and its input.
Structural *representation power*	The part of an *object system structure* that is represented by a *model*. Note 1. *Object system structure* is a concept of the problem ontology, described in Chapter 8.
Model objective	The purpose of a model within its *representation power*. Note 1. Not to be confuse with the modelling project objective, described in Chapter 6, the *define objectives* (task) Note 2. Not to be confused with the *objective function*, which is a mathematical defined goal.
Model objective description	Description of the *model objective*, given a specific *model objective type*.
Model objective types	The perspective aimed by a model, being *descriptive*, *prescriptive* or *explanatory*.

[1] The concept of *mathematical modelling paradigm* has been defined in Chapter 6 and Appendix D.

Descriptive *model objective*	A *model objective type* that is *descriptive* includes: • Traces what actually happens during a process; • Takes the point of view of an external observer who looks at the way a process has been performed and determines the improvements that have to be made to make it perform more effectively or efficiently. Note 1. Source: Wikipedia.
Prescriptive *model objective*	A *model objective type* that is *prescriptive* includes: • Defines desired processes and how they should/could/might be performed; • Lays down rules, guidelines, and behavior patterns, which, if followed, would lead to the desired process performance. They range from strict enforcement to flexible guidance. Note 1. Source: Wikipedia.
Explanatory *model objective*	A *model objective type* that is *explanatory* includes: • Provides explanations about the rationale of processes; • Explores and evaluates several possible courses of action based on rational arguments; • Establishes an explicit link between processes and the requirements that they are to fulfill. Note 1. Source: Wikipedia.
Base *model*	The base model is a model capable of accounting for all input-output behavior of the real system. In other words, it is valid in all the allowable experimental frames. In any realistic modelling and simulation area, the base model can never be fully known, although certain aspects of its description may be expected as known. Since the base model provides a complete explanation of the behavior of a real system, it may be expected to comprise many, many components and interactions. Note 1. Source: Zeigler (1976). Note 2. A base model is the perfect representation of all observed and unobserved entities and relations in the *object system* (see Chapter 8). Note 3. Zeigler (1976) further defines a 'lumped model', which can best be compared with what here is called a *model* and which reflects obj*ect system entities* and *object system relations* that are relevant for the *problem* at hand. This should not be (not to be confused with Cellier's (1991) 'lumped parameter model'.
Optimization *model*	A *mathematical model* belonging to the *optimization modelling paradigm*. Note 1. The *optimization modelling paradigm* has been defined in Chapter 6 and Appendix D.
Linear programming model	A *mathematical model* belonging to the *linear programming modelling paradigm*. Note 1. The *linear programming modelling paradigm* has been defined in Chapter 6 and Appendix D.
Non-linear programming model	A *mathematical model* belonging to the *non-linear programming modelling paradigm*. Note 1. The *non-linear programming modelling paradigm* has been defined in Chapter 6 and Appendix D.

Simulation *model*	A mathematical model belonging to the *simulation modelling paradigm*. Note 1. The *simulation modelling paradigm* has been defined in Chapter 6 and Appendix D. Note 2. In fact
Continuous *simulation model*	A *simulation model* belonging to the *continuous simulation modelling paradigm*. Note 1. The *continuous simulation modelling paradigm* has been defined in Chapter 6 and Appendix D.
Discrete *simulation model*	A *simulation model* belonging to the *discrete simulation modelling paradigm*. Note 1. The *discrete simulation modelling paradigm* has been defined in Chapter 6 and Appendix D.
Agent based *model*	A *simulation model* belonging to the *agent based simulation modelling paradigm*. Note 1. The a*gent based simulation modelling paradigm* has been defined in Chapter 6 and Appendix D.

G.2 Ontological layer 2: mathematical model knowledge

Ontological layer 2 contains *mathematical model knowledge* and is filled with *concepts* and the *relations* connecting the concepts, which are common for all *mathematical models*, i.e. belonging to all *mathematical modelling paradigms*[2]. The terminology, i.e. *concepts* and *relations* in this layer should be instrumental for discussing the various instances of *ontological layers 3*. Each of the latter will contain terminology, i.e. *concepts* and *relations* for the *simulation modelling paradigm*. In this section all concepts used in that ontology are defined in Table G-2. This *ontological layer* contains all concepts depicted in the structure diagram of Figure 9-4 of Chapter 9.

Table G-2. Concepts of *ontological layer 2* (mathematical model knowledge). All terms in *italic* are defined components of the *meta-ontology* (Chapter 5).

Concepts	Meaning
Model mode	The format in which a model can occur, being *conceptual model*, *mathematical model, computer model*, *site specific computer model*.
Conceptual *model mode*	*Model mode*, in which the *model structure*, i.e. *model quantities* such as *state variables* and physical process describing model relations is represented in one or more of the following formats: text *conceptual model diagram* and a mathematical description, i.e. the *mathematical model*. Note 1. Sometimes the *conceptual mode* of a *model* includes the *mathematical mode* of a *model*, i.e. all *equations*, *parameters*, *initial conditions* and *boundary conditions*. Note 2. The conceptual model reflects the chosen abstraction and aggregation *layer* of a model. Note 3. Refsgaard and Henriksen (2004) state that the *conceptual model* constitutes the scientific hypothesis or theory that is assumed for the particular modelling study.
Conceptual model diagram	Diagram depicting the qualitative structure of the *conceptual model*. Note 1. A *conceptual model diagram* can have different formats and layouts, depending on the *model paradigm*, e.g. a cause-effect diagram, a stocks and flows diagram, etc.; an example of a conceptual model and the mathematical model based on it is given in Chapter 9, section 9.3.5.
Mathematical *model mode*	The model mode of a mathematical model.

[2] The concept of *mathematical modelling paradigm* has been defined in Chapter 6 and Appendix D.

Computer *model mode*	A *mathematical model* implemented in a computer program. Note 1. Refsgaard and Henriksen (2004) call this a 'model code'. Note 2. Often abbreviated to *computer model*. Note 3. Often used synonyms: model application, model, simulation model, etc.
Site specific *computer model*	A *computer model* including *object system input* and some *numerical model solver*, suited for the *model paradigm* to which the *model* belongs. Note 1. Refsgaard and Henriksen (2004) call this 'model' or 'site specific model'.
Model scope	The circumstances or conditions under which a model has documented predictive capabilities. Note 1. Source: HarmoniQuA glossary: http://www.harmoni-ca.info/Registered_Users/Glossary/index.php.
Theoretical *model scope*	The set of circumstances where the model applies in theory. Note 1. Source: Edmonds (2005).
Practical *model scope*	The set of circumstances where the model applies in practice. Note 1. Source: Edmonds (2005). Note 2. Newton's laws of physics presumably apply to all macroscopic situations with a high degree of accuracy, but in practice are very difficult to apply unless the objects and forces involved are discrete and identifiable.
Scope of *conceptual model*	Prescribed conditions for which the *conceptual model* has been tested, i.e. compared with the *object system* to the extent possible and judged suitable for use by *model confirmation*. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) use 'reality' instead of *object system*. Note 3. *Model confirmation* is defined in Appendix D.
Scope of *computer model*	Prescribed conditions for which the *computer model* has been tested, i.e. compared with the analytical solutions, other *computer models* or similar to the extent possible and judged suitable for use by *computer model verification*. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) use 'model code' instead of *computer model*. Note 3. *Computer model verification* is defined in Appendix D.
Scope of *site specific computer model*	Prescribed conditions for which the *site specific computer model* has been tested, i.e. compared with the *object system* to the extent possible and judged suitable for use by *model validation*. Note 1. Adapted from Refsgaard and Henriksen, 2004. Note 2. Refsgaard and Henriksen (2004) use 'site specific model' instead of *site specific computer model*. Note 3. *Model validation confirmation* is defined in Appendix D.
Numerical *model* solver	A numerical algorithm suited to solve numerically (i.e. using computers) a *mathematical model*. *Model solvers* are typical for a *model paradigm*.
Objective function	The *objective function*, $F(x)$ is the mathematical function that has to be optimized (i.e. minimized or maximized) by varying a n-vector x of *model inputs* (e.g. *parameters* in case of *simulation modelling* or *decision variables* in case of *optimization modelling*. Optimization is called constrained in case x is constrained. Note 1. The terms *objective function* and the related terms *norm* and *criterion* are briefly discussed in Appendix A in the context of *simulation modelling*. Note 2. Examples of optimization in OR (i.e. operations research) and simulation modelling can be found in Appendix A.

Model assumption	A statement on how a *model* simplifies the *functional knowledge*, the *structural knowledge* and the *process knowledge* in the *object system*. Note 1. *Object system* has been defined in Chapter 8. Note 2. An example is using the room temperature measured at one point in a room as representative for the whole room.
Model component	Part of a model that is a *model* in itself.
Atomic *model*	Smallest *model component*, which is a *model* in itself. Note 1. Typically a single equation.
Composite *model*	Group of *model components*.
Sub*model*	Group of *model components* and therefore a *composite model component*.
I/O interface	The collection of interfaces that *atomic model components* use to communicate with each other. *Model inputs* are the data received and *model output* the data sent from it.
Model quantity	The representation of an *object system entity* in a *model*. Note 1. The American Heritage Dictionary of the English Language, Fourth Edition, defines quantity as ‘The measurable, countable or comparable property of a thing.’ Note 2. *Quantity* can be divided in ‘magnitude’ and ‘multitude’ (number). Note 3. A proper synonym for *model quantity* would be *model entity*, but *quantity* emphasizes that it has a size and a *unit.*
Unit	A precisely specified quantity in terms of which the magnitudes of other quantities of the same kind can be stated. Note 1. Example: units in the SI (Système International d'Unités), the metric system. Note 2. *Unit* is a property of *model quantities*.
Model input	*Model quantity*, which is input of the model. Note 1. Adapted from ‘input variable’ in NEN-6260-1 (2002). Note 2. In simulation models (layer 2), typically an *observable variable*.
Model output	*Model quantity*, which is stored in the output. Note 1. Adapted from ‘input variable’ in NEN-6260-1 (2002). Note 2. In simulation models (layer 2), typically an observable variable.
Model variable type	*Model quantity*, which is a model representation of object system entities in the problem ontology. Note 1. The problem ontology is discussed in Chapter 8. Note 2. A *model variable type* can be compared with its associated *object system entity*.
Model parameter type	*Model quantity*, which is assumed constant during a (simulation) run, but which does not have to be accurately and precisely known Note 1. Source: NEN-6260-1 (2002). Note 2. Synonym coefficient.
Model function type	*Model quantity*, which associates every element (input, argument) of a certain set of numbers or other objects to a corresponding element (output, result) in some other set. Note 1. Adapted from: Wikipedia.
Model operand	One of the inputs (arguments) of a *model operator*. Note 1. Source: Wikipedia. Note 2. For instance, in ‘3+6=9’ is ‘+’ the *operator* and ‘3’ and ‘6’ are the *operands*.

(*Model*) operator, (*model*) operation	1. A function, especially one from a set to itself, such as differentiation of a differentiable function or rotation of a vector. 2. Operation a procedure for generating a value from one or more other values (the *operands*; the value for any particular operands is unique). *Operator*: a symbol representing an operation. 3. In Mathematics a. A mathematical process, as addition, multiplication, or differentiation. b. The action of applying a mathematical process to a quantity or quantities. Note 1. Source of meaning 1: The American Heritage Dictionary of the English Language, Fourth Edition. Note 2. Source of meaning 2: Wiktionary http://en.wiktionary.org/wiki/Operation Note 3. Source of meaning 3: http://www.infoplease.com/ipd/A0567745.html Note 4.Examples of *operators*: • The differential *operator*: d/dt • The indefinite integral *operator*: $\int_0^t$ • '3+6=9' is '+' the *operator* and '3' and '6' are the *operands*.
Unary *operation*	*Operation* with one *operand*. Note 1. Source: Wikipedia. Note 2. With a single operand means with a single input. Note 3. Example: logical negation (-), squaring on real numbers (√).
Binary *operation*	*Operation* with two *operands*. Note 1. Source: Wikipedia. Note 2. Example: addition (+), subtraction (), multiplication (*), division (/) and exponentiation (^).
Arithmetic *operator*	One of the following operators: *addition*, *subtraction*, *multiplication* and *division*.
Operator symbol	A notation that denotes the type of mathematical object that is known as an *operation*.
Operation notation	Systematic way of writing operators and their arguments. Note 1. Source: Wikipedia. Note 2. There are five types of operation notations: prefix, postfix, infix, juxtaposition and with superscripts and subscripts. Note 3. In models typically an infix notation is used.
Differential	*Binary operation*, which is an infinitesimal change in the value of a *function*. The *derivative* and integral are defined in terms of a limit of a *differential*. Note 1. Symbol D or d/d(independent variable). Note 2. Source Wikipedia.
Indefinite integral	*Binary operation*, which is the set of all antiderivatives of a given function *f*. Note 1. Symbol: $\int_0^t$. Note 2. Source Wikipedia.
Addition	*Binary operation*, which is the basic *arithmetic operations*. In its simplest form, *addition* combines two numbers, the *addends* or *terms*, into a single number, the *sum*. Note 1. Source: Wikipedia.

Subtraction	*Binary operation*, which is one of the four basic *arithmetic operations*; it is essentially the opposite of *addition*. *Subtraction* is denoted by an minus sign (-) in *infix notation*. Note 1. Source: Wikipedia.
Multiplication	*Binary operation*, which is an *arithmetic operation* that is the inverse of *division*, and in elementary arithmetic, can be interpreted as repeated *addition*. In its simplest form, *multiplication* is the sum (*addition*) of a list of identical numbers. Note 1. Source: Wikipedia.
Division	*Binary operation*, which is an *arithmetic operation* that is the inverse of *multiplication*. Note 1. Source: Wikipedia.
Exponentiation	*Binary operation*, which is repeated *multiplication*. Note 1. Source: Wikipedia.
Model expression	Any mathematical calculation or formula combination numbers and/or *variables* using sums, differences, products, quotients (including fractions), exponents, roots, logarithms, trig functions, parenthesis, brackets, functions, or other mathematical operations. *Expressions* may not contain the equal sign or any type of inequality. Note 1. Adapted from www.mathwords.com/a_to_z.htm. Note 2. *Model expressions* should be 'well-formed', similar to 'well formed' expressions, defined by Russell and Norvig (1995) as 'sentences that have all their variables properly introduced.
Model inequality	A mathematical sentence built from expressions using one or more operators of the types: <, >, ≤, ≥. Note 1. Adapted from www.mathwords.com/a_to_z.htm. Note 2. A mathematical sentence is here defined as (*atomic*) *model component*.
Model equation	1. A mathematical sentence built from expressions using one or more equal signs (=). 2. In models equations typically assign a value to a *variable* by defining the *variable* with an *expression*. Note 1.Meaning 1 is Adapted from www.mathwords.com/a_to_z.htm. Note 2. A 'mathematical sentence' is here defined as (*atomic*) *model component*. Note 3. *Model equations* are often in the form of what is called an assignment (statement) in computer science; notation: *model variable type* = *model expression*.

G.3 Ontological layer 3: (continuous) simulation model knowledge

Ontological layer 3 is filled with *concepts* and the *relations* connecting them, which are specific for a single *mathematical model paradigm*. The mathematical modelling paradigm defined in this section is *(continuous) simulation model knowledge*. The terminology, i.e. *concepts* and *relations* in this layer should be instrumental to define and discuss simulation models for a class of continuous simulation model, of which a single instance (i.e. continuous simulation models for generic bivalve ecophysiological models) is defined in *ontological layer 4* (see Chapter 9, section 9.3.6). Some of the concepts are derived from the development of Smart, Simulation and Modelling Assistant for Research and Training (Kramer and Scholten, 2001). This *ontological layer 3* contains all concepts depicted in the structure diagram of Figure 9-5 of Chapter 9.

Table G-3. Concepts of *ontological layer 3* (continuous) simulation model knowledge). All terms in italic are defined concepts of the *meta-ontology* (Chapter 5) or of *ontological 1* and *2*.

Concepts	Meaning
Observable variable	*Model variable type*, which is a model representation of an *observable object system entity*. Note 1. *Observable object system entity* is defined in the problem ontology of Chapter 8.
Not observable variable	*Model variable type*, which is a model representation of a *not observable object system entity*. Note 1. *Not observable object system entity* is defined in the problem ontology of Chapter 8.
Observed variable	*Model variable type*, which is a model representation of an *observed object system entity*. Note 1. *Observed object system entity* is defined in the problem ontology of Chapter 8.
Not observed variable	*Model variable type*, which is a model representation of a *not observed object system entity*. Note 1. *Not observed object system entity* is defined in the problem ontology of Chapter 8.
State variable	*Model variable type* used to describe the state of the *process/object system*. Note 1. Source: NEN-6260-1 (2002). Note 2. State variables are described with (partial) differential equations; the value of a state variable at a specific place and point in time is calculated based on the (partial) differential equations and its value on one or more points in time and other places. Note 3. A *state variable* is also referred to as a dependent *variable*, i.e. dependent of the *independent variable*. In dx/dt is 'x' the dependent *variable* (and state variable) and 't' the *independent variable*. Note 2. 'Stock' is a synonym for *state variable*.
Independent *variable*	*Model variable type* used to describe changes in the *dependent variables*. In a dynamical system time is an *independent variable* and in a spatially articulated model 1, 2 or 3 spatial dimensions are *independent variables*. In other words an *independent variable* is the variable to which an *dependent variable* is differentiated in a differential equation Note 1. Source: NEN-6260-1 (2002).
Auxiliary *variable*	*Model variable type* that is no *state variable* (i.e. *variable* which value does not depend on its value on a previous value of the *independent variable* (e.g. not of the value at a previous point in time) Note 1. Source: NEN-6260-1 (2002).
Dependent *variable*	See *state variable* and *independent variable*. Note 1. Source: NEN-6260-1 (2002).
Constant *parameter*	*Model parameter type* that does not change in time, like nature constants. Note 1. Adapted from 'input variable' in NEN-6260-1 (2002).
Observed *parameter*	*Model parameter type* that has been observed or measured in the field, laboratory experiment or derived from literature, which uncertainty can be expressed as a known or unknown statistical distribution.

Decision *parameter*	*Model parameter type* used to calculate the effects of (human) intervention in the system with the model Note 1. Adapted from 'input variable' in NEN-6260-1 (2002). Note 2. Typically *model input*. Note 3. Often referred to as 'decision variable'.
Calculated *parameter*	*Model parameter type* calculated from other quantities, e.g. *model variable type* and *model parameter type*.
Free *parameter*	*Model parameter type*, which is not an *observed parameter* or a *calculated parameter*, but which value can freely be chosen. Note 1. Its role is comparable to that of a decision parameter, but without a clear link to human intervention.
Boundary condition	Boundary conditions are a set of values needed to solve the differential equation unambiguously. Boundary conditions are typically values for state variables at the spatial boundaries. Note 1. Source: NEN-6260-1 (2002).
Initial condition, initial value	The value of a state variable at the begin of a simulation.
Time series	Set of values of a *model function type*, which is connected to the corresponding *observed object system entity*, observed at sequential points in time. Note 1. Observed object system entity is defined in the problem ontology of Chapter 8.
Spatial series	Set of values of a *model function type*, which is connected to the corresponding *observed object system entity*, observed at various (spatial) locations. Note 1. Observed object system entity is defined in the problem ontology of Chapter 8.
Spatio-temporal series	Set of values of a *model function type*, which is connected to the corresponding *observed object system entity*, observed at various (spatial) locations and at sequential points in time. Note 1. Observed object system entity is defined in the problem ontology of Chapter 8.
Tabular function	Set of values of *model function type*, which value is calculated from a table of values by (some form of) interpolation.
Basic function	A set of standard functions, that also of *model function type* including goniometric functions (sin, cos, etc.) logarithmic functions and others. Note 1. The following are regarded as basic functions: *sine*, *cosine*, *tangent*, *arc sin*, *arc cosine*, *arc tangent*, *constant pi*, *square root*, *exp* (i.e. e^y with e base of natural logarithm), *power* (i.e. x^y), *ln* (i.e. natural logarithm, log10 (logarithm base 10), *absolute value*, *min2* (i.e. minimum of 2 arguments), *max2* (i.e. maximum of 2 arguments), *limit*, *dead space* Note 2. The basic function *limit*(*variable*, *low_limit*, *high_limit*) is defined as follows: if *low_limit≤variable≤high_limit* then *variable=variable* other wise if *variable<low_limit* then *variable* = *low_limit* if *variable>high_limit* then *variable=high_limit*) Note 3. The basic function *dead space*(*variable*, *low_limit*, *high_limit*);is defined as follows: If *low_limit≤variable≤high_limit* then *variable=0* other wise, if *variable<low_limit* then *variable* = variable–*low_limit* if *variable>high_limit* then *variable*=variable–*high_limit*
Other function	Other, user defined mathematical *functions*.
Derivative	The instantaneous rate of change of a *function*. Note 1. Source: Wikipedia.

Partial *derivative*	The derivative of a *function* of several variables with respect to one of those *variables* with the others held constant (as opposed to the total *derivative*, in which all *variables* are allowed to vary). Note 1. Source: Wikipedia; The *partial derivative* of a *function f* with respect to the *variable x* is represented as δf/δx..
Differential *equation*	An *equation* in which the *derivatives* of a *function* appear as *variables* Note 1. Source: Wikipedia. Note 2. Many of the fundamental laws of physics, chemistry, biology and economics can be formulated as differential equations. Note 3. *Equations* in *models* are often in the form of what is called an assignment (statement) in computer science; notation: *state variable* = *rate variable*.
Ordinary *differential equation*	A *differential equation* which only contains functions of one *independent variable*, and *derivatives* in that variable. Note 1. Source: Wikipedia. Note 2. Typically it consists of the differential operator applied on a state variable, the equal sign and a rate variable: d(*state variable*)/dt = *rate variable*
Partial *differential equation*	A *differential equation* which contains *functions* of multiple *independent variables* and their *partial derivatives*. Note 1. Source: Wikipedia.
Algebraic equation	An algebraic equation is an mathematical equation of the form $P = 0$ where P is an algebraic expression, e.g. a (possibly multivariate) algebraic polynomial. For example $x^2 + 3xy - 4y^2 + 1 = 0$ Note 1. Adapted from Wikipedia. Note 2. *Algebraic equations* in *models* are often in the form of what is called an assignment (statement) in computer science; notation: *auxiliary variable* = *expression*.
Algebraic *expression*	An *expression* is algebraic if it involves a finite combination of numbers and *variables* and algebraic *operations* (addition, subtraction, multiplication, division, raising to a power and extracting a root).
Conceptual *simulation model*	*Simulation model* in the *conceptual model mode*. Note 1. Often a *conceptual simulation model* describes the state variables and the flow variables.
Mathematical *simulation model*	*Simulation model* in the *mathematical model mode*. Note 1. In its most concise form it is the set of all differential equations. Note 2. For practical reasons (better understanding) a right hand part of the differential equation, i.e. the *rate variable*, is often split in more *flow variables*.
Computer *simulation model*	*Simulation model* in the *computer model mode*.
Site specific *computer simulation model*	A *computer simulation model* with its problem and object system related inputs. Note 1. In a *site specific computer simulation model* there are links to the ontology describing the *problem* and the associated *object system*, which is described in Chapter 8.
Rate variable	An *expression* defining the change in a *state variable*, consisting of the summation of all inflows (positive) and outflows (negative) of all outflows of a state variable. Note 1. A rate variable is typically defined by an expression.
Flow variable	*Model variable type* representing an exchange between two state variables. Note 1. Arrows in *conceptual model diagrams* often represent *flow variables*.

Inflow	*Flow variable* increasing the value of a state variable.
Outflow	*Flow variable* decreasing the value of a state variable.
Simulation, (simulation) run	The use of a *mathematical model* as an experimental vehicle to answer questions about a *problem* and associated *object system*. Note 1. *Problem* and *object system* are defined in Chapter 8. Note 2. This term is not a part of the model, but of what can be done with a model.
Run options	Set of selections that determine the *simulation run*, including start of simulation, end of simulation, integration methods, etc. Note 1. *Run options* are a part of a *model experiment*. Note 2. This term is not a part of the *model*, but of what can be done with a *model*.
Model experiment	The assemblage of *experimental frame*, *run options* and *model output*. Note 1. This term is not a part of the model, but of what can be done with a model.
Experimental frame	A limited set of circumstances under which the real system is to be observed or experimented with Note 1. Source: Zeigler (1976). Note 2. Zeigler (1976) postulates: The real system observed within the experimental frame *E* is structurally characterized by a model denoted by *B/E* and called the *base model* in *E*. Note 3. This term is not a part of the *model*, but links a (generic) model with real system properties and makes so the model specific for that system and time period. Note 4. An *experimental frame* is a part of a *model experiment*.
Model scenario	Set of 'model inputs', aiming at achieving a specific *model* result. Note 1. 'Model input' can consist of *model parameters*, *model functions* and *run options*. Note 2. This term is not a part of the model, but of what can be done with a model.

G.4 Ontological layer 4: simulation model knowledge for application domains, instantiated for bivalve ecology models

Ontological layer 4 (simulation model knowledge for application domains, instantiated for bivalve ecology models) is more specialized than *ontological layer 3* ((continuous) simulation model knowledge), as it contains scientific, very detailed factual knowledge. This *ontological layer* contains all concepts depicted in the structure diagram of Figure 9-8 of Chapter 9 and all concepts, defined in *ontological layer 3*, are instantiated in the mathematical model EMMY (see Table G-4, Table G-5, Table G-6, Table G-6, Table G-7 and Table G-8). This mathematical has subsequently be implemented in a *computer model* and has been *solved numerically* with a proper *solver*. This has been discussed in Chapter 9, section 9.3.6.2.

Table G-4. *State variables* of the *continuous simulation model* for bivalve ecophysiology. 'C' refers to carbon, 'N' to nitrogen and 'DW' to dry weight. The *differential equations* with nitrogen are irrelevant for cockles and therefore they will be included in EMMY, but not in COCO. The knowledge in this table is explained in Chapter 8 (especially section 8.3.4). The *rate variables* are defined in Table G-6.

Derivatives of state variable		Rate variables	Units	Meaning
d(SomC)/dt	=	SomCpos – SomCneg	[gC]	C in *somatic compartment*
d(SomN)/dt	=	SomNpos – SomNneg – SomGamN	[gN]	N in *somatic compartment*
d(StorC)/dt	=	StorCpos – StorCneg – StorGamC	[gC]	C in *storage compartment*
d(GamC)/dt	=	StorGamC – SpawningC	[gC]	C in *reproductive compartment*
d(GamC)/dt	=	SomGamN – SpawningN	[gN]	N in *reproductive compartment*
d(OSW)/dt	=	OSWR	[gDW]	DW in *shell compartment*

Table G-5. Required time series as input for the model. All time series are assumed to be averaged over a time unit, i.e. a day.

Timeseries	Unit	Meaning	Data type
Temp(t)	[°C]	Water temperature	From database
TPM(t)	[$g.m^{-3}$]	*Total particulate matter*	From database
POM(t)	[$g.m^{-3}$]	*Particulate organic matter*	Calculated from POC

POC(t)	[gC.m^{-3}]	*Particulate organic carbon*	From database
PON(t)	[gN.m^{-3}]	*Particulate organic nitrogen*	Calculated function
CHLF(t)	[mgCHLF.m^{-3}]	*Chlorophyll*	From database
Phaeo(t)	[cells.ml^{-1}]	*Phaeocystis colonies*	From database

Table G-6. Auxiliary variables (including rate variables), their mathematical definitions and units.

Variable	=	Expression		Meaning
		Food related processes		
TPM	=	Temp(t) · TPMbot	[g.m^{-3}]	*Total Particulate Matter* (*seston*)
POC	=	POC(t) · POCbot	[gC.m^{-3}]	*Particulate Organic Carbon*
PON	=	PON(t) · POC	[gN.m^{-3}]	*Particulate Organic Nitrogen*
POM	=	POM(t) · POMbot	[gDW.m^{-3}]	*Particulate Organic Matter*
fytoPOM	=	CHLF(t) · CHLFbot · fCCHL · POCtoPOM	[gDW.m^{-3}]	*POM* in *chlorophyll*
detPOM	=	POM – fytoPOM	[gDW.m^{-3}]	*Detritus* part of *POM*
labDet	=	max(0,detPOM · fLabDet)	[gDW.m^{-3}] [gDW.m^{-3}]	*Labile detritus*
Food	=	fytoPOM + LabDet	[gDW.m^{-3}]	Total edible *food*
fPOMses	=	min(1,Food /TPM)	[-]	*POM* fraction of *seston*
CDWfood	=	POC / POM	[gC g^{-1} DW food]	Fraction carbon in *food*
NDWfood	=	PON / POM	[gN g^{-1} DW food]	Fraction nitrogen in *food*
		Filtration		
Tcr	=	Q10cr$^{(temp(t)-10)/10}$	[-]	Temperature function of *clearance*
CR	=	Tcr · submerged · Acr · Wtot^{Bcr}	[m^{-3}.d^{-1}]	*Clearance* rate; *clearance* occurs only during *submersion*

CDsest	=	if (Asest < TPM < Bsest) then
		(Cdmax · (TPM-Asest)/(Bsest-Asest))
		if (TPM < Asest) then 0
		if (TPM > Bsest) then CDmax
			[-]	*Clearance depression* due to high *seston* concentration

Ncol	=	phaeo(t) / Ncellcol	[col.ml^{-1}]	*Phaeocystis-colonies* concentration
TCS	=	Ncol · (PhDiam/2)^2 · π	[mm^2.ml^{-1}]	The sum of all (total) cross sections of *Phaeocystis colonies*

CDphaeo	=	if (Aphaeo < TCS < Bphaeo) then
		(Cdmax · (TPM-Aphaeo)/(Bphaeo-Aphaeo))
		if (TCS < Aphaeo) then 0
		if (TCS > Aphaeo) then CDmax
			[-]	*Clearance depression* due to high concentration of *Phaeocystis-colonies*

CDpom	=	if (fPOMses < Apom) then 0
		if (fPOMses < Apom) then
		(Cdmax · (fPOMses - Apom) / (1 – Asest))
			[-]	*Clearance depression* caused by 'pure' *phytoplankton* diet

CD	=	max(CDsest,CDphaeo,CDpom)	[-]	Effective *clearance depression*
FR	=	CR · TPM	[g.d^{-1}]	*Filtration rate* of 1 mussel
OFR	=	FR · fPOMses	[gDW.d^{-1}]	Organic *filtration rate*
		Pre-ingestive selection and ingestion		
GC	=	Agc · Wtot^{Bgc}	[mm^3]	*Gut content*
fPOMing	=	fPOMses^{Bses}	[-]	*POM* fraction of *ingested seston*
GPT	=	fPOMing · (GPTmin – GPTmax) + GPTmax	[h]	*Gut passage time*
IRmax	=	(GC · SpSesMass · 24) / GPT	[g.d^{-1}]	Maximum *ingestion rate* of 1 mussel
IR	=	if (FR < Irmax) then FR else IRmax	[g.d^{-1}]	*Ingestion rate* of 1 mussel
RejR	=	max(0,FR-IR)	[g.d^{-1}]	*Rejection rate* of 1 mussel
OIR	=	min(OFR,fPOMing · IR)	[gDW.d^{-1}]	*Organic ingestion rate*
fPOMpsf	=	if (RejR ≤ 0) then ((OFR – OIR) / RejR) else 0	[-]	*POM* fraction in *pseudofaeces*
SE	=	1 – (fPOMpsf / fPOMses)	[-]	*Selection efficiency*: *food* particles/*seston*
Mucus	=	MucusCoef · RejR	[gDW.d^{-1}]	*Mucus production* depending on *rejection rate* (*pseudofaeces*)
MucusC	=	Mucus · CDWfrac	[gC.d^{-1}]	Carbon in *mucus*
		Absorption		
AEC	=	AEmax · (1. - exp(-BetaC·GPT))	[-]	*Absorption efficiency* for carbon
AEN	=	AEmax · (1. - exp(-BetaN·GPT))	[-]	*Absorption efficiency* for nitrogen
ARC	=	AEC · OIR · CDWfood	[gC.d^{-1}]	*Absorption rate* carbon
ARN	=	AEN · OIR · NDWfood	[gN.d^{-1}]	*Absorption rate* nitrogen
FaecR	=	max(0.,IR - (ARC /CDWfood))	[g.d^{-1}]	*Faeces rate* (IR - *absorbed* material)
		Respiration and excretion		
Tres	=	Q10res$^{(temp(t)-10)/10}$	[-]	Temperature function *respiration*

ResRbas	=	Tres · Ares · Wtot^{Bres}	[gC.d^{-1}]	Bodyweight depending *respiration rate*
ResRout	=	ResAbsC · OIR · submerged	[gC.d^{-1}]	*Routine respiration* depending of *absorption*; this occurs only during *submersion*
ResR	=	ResRbas + ResRout	[gC.d^{-1}]	Total effective *respiration rate*
ResOperH	=	ResR · (3200/24)	[mgO$_2$.h^{-1}]	*Respiration rate* oxygen
ExcRbas	=	Aexcr · Wtot^{Bexcr}	[gN.d^{-1}]	Basic *excretion* of nitrogen as NH$_4$
ExcRcor	=	if (TotCN > Cnratmin) then 0 else ((NDWfrac/CDWfrac) · (ResR + MucusC))	[gN.d^{-1}]	Correction for C-budget in terms of nitrogen
ExcR	=	ExcRbas + ExcRcor	[gN.d^{-1}]	Total nitrogen *excretion* as NH$_4$
ExcRperH	=	ExcR · (1000000/24)	[µgN.h^{-1}]	*Excretion* per hour
		Allocation		
NetProdC	=	ARC - MucusC - ResR	[gC.d^{-1}]	Net production carbon
NetProdN	=	ARN – ExcR	[gN.d^{-1}]	Net production nitrogen
ToStor	=	1/(1 + KStor · exp(-KStorExp·time))	[-]	Fraction of net production to storage compartment[3]
ToOSW	=	OSWperSOM · (1 - ToStor)	[-]	Fraction of net production to *organic shell compartment*
ToSom	=	(1 - OSWperSOM) · (1 - ToStor)	[-]	Fraction of net production to *somatic compartment*
SomCpos	=	if (NetProdC > 0) then (ToSom · NetProdC)	[gC.d^{-1}]	Carbon *flow* from net production to *somatic compartment*
StorCpos	=	if (NetProdC > 0) then (ToStor · NetProdC)	[gC.d^{-1}]	Carbon *flow* from net production to storage compartment
OSWCR	=	if (NetProdC > 0) then (ToOSW · NetProdC)	[gC.d^{-1}]	*Rate variable* carbon in *organic shell compartment*

StorCneg	=	if (StorC > (–NetProdC)) then (- NetProdC)
		else
		if (SomC > (–NetProdC – StorC)) then StorC
			[gC.d^{-1}]	Carbon *flow* from *storage compartment* to *respiration* (ResR) and *mucus production* (MucusC)

SomCneg	=	if (SomC > (–NetProdC – StorC) then (– NetProdC – StorC)	[gC.d^{-1}]	Carbon *flow* from *somatic compartment* to *respiration* (ResR) and *mucus production* (MucusC)
OSWR	=	OSWCR / CDWfrac	[gC.d^{-1}]	*Rate variable* carbon in *organic shell compartment* (OSWC), expressed as *dry weight*
SomNpos	=	if (NetProdN > 0) then (NetProdN · (ToSom + ToStor))	[gN.d^{-1}]	Nitrogen *flow* from net production to *somatic compartment*
OSWNR	=	if (NetProdN > 0) then (ToOSW · NetProdN)	[gN.d^{-1}]	*Rate variable* nitrogen in *organic shell compartment*
SomNneg	=	if (NetProdN+SomN > 0) then (–NetProdN)	[gN.d^{-1}]	Nitrogen *flow* from *somatic compartment* to *excretion* (ExcR)
		Reproduction		
GamGen	=	if (StorFrac > GamGenTrig) then true	[true/false]	Button to switch *gametogenesis* on/off
StorGamC	=	if ((GamGen) and (StorFrac > StorCtresh)) then (GamRate · StorC) else 0	[gC.d^{-1}]	Carbon *flow* from *storage compartment* to *reproductive compartment* (*gametes*)
SomGamN	=	if ((GamGen) and (StorFrac > StorCtresh)) then (NDgam · StorGamC / CDWfrac) else 0	[gN.d^{-1}]	Nitrogen *flow* from *somatic compartment* to *reproductive compartment* ((*gametes*))
SpawningC	=	if ((temp(t) > TempGam)) then (GamC + StorGamC)	[gC.d^{-1}]	Carbon in *spawning*
SpawningN	=	if (temp(t) > TempGam) then (GamN + SomGamN)	[gN.d^{-1}]	Nitrogen in *spawning*
		Extra output variables		
Wsom	=	SomC / CDWfrac	[gDW]	*Dry weight* of *somatic compartment*
Wstor	=	StorC / CDWfrac	[gDW]	*Dry weight* of *storage compartment*
Wgam	=	GamC / CDWfrac	[gDW]	*Dry weight* of *reproductive compartment*
Wtot	=	Wsom + Wstor + Wgam	[gDW]	Total *dry weight* of mussel flesh
GamFrac	=	Wgam / Wtot	[-]	*Reproductive compartment* as fraction of total *dry weight*

[3] A better formula for ToStor uses bodysize (Wtot) as measure for age, leading to:
ToStor = (LowFr · HighFr)/(LowFr + (HighFr - LowFr) · $e^{-KstorExp \cdot Wtot}$) with parameter values: LowFr=0.01, HighFr=0.4 and KstorExp=4. A bivalve of 1 gDW will allocate 23 % of NetProdC and larger individuals up to 40 % (equal to HighFr).

StorFrac	=	Wstor / Wtot	[-]	*Storage compartment* as fraction of total *dry weight*
SomFrac	=	Wsom / Wtot	[-]	*Somatic compartment* as fraction of total *dry weight*
TotCN	=	(SomC + StorC + GamC) / (SomN + GamN)	[gC.gN-1]	Total CN-ratio
DSW	=	OSW · OSWtoDSW	[g]	*Dry Shell Weight*
Lshell	=	Ashell · DSW^{Bshell}	[mm]	*Shell length*
TWW	=	dShell · $LShell^{Lcubic}$	[gWW]	Total *wet weight* (shell + flesh)
CFW	=	Wtot · DryToFlesh	[g]	Cooked *flesh weight*
PercFlesh	=	(CFW / TWW) · 100	[%]	Percentage flesh
Fecundity	=	Wspawn/Wtot	[-]	Fraction *dry weight* spawned

Table G-7. List of parameters treated as constants in calibration.

Parameter	Value	Units	Reference	Comment
APHAEO	0.5	[$mm^2.ml^{-1}$]	Smaal & Twisk (1997)	Lower threshold Phaeocystis colony surface for clearance depression
APOM	0.6	[-]		fPOMses threshold for clearance depression
ASEST	35	[$g.m^{-3}$]	Klepper, 1989, mod.	Lower threshold seston concentration for clearance depression
BETAC	0.5	[h^{-1}]	Willows, 1992, mod.	Energy absorption rate carbon
BETAN	0.5	[h^{-1}]	Willows, 1992, mod.	Energy absorption rate nitrogen
BPHAEO	6.5	[$mm^2.ml^{-1}$]	Smaal & Twisk (1997)	Upper threshold Phaeocystis colony surface for clearance depression
BSEST	150	[$g.m^{-3}$]	Prins et al. (1991)	Upper threshold seston concentration for clearance depression
CDMAX	0.8	[-]		Maximum clearance depression (mussel beds = 0.8)
CDMAXPOM	0.87	[-]		Maximum clearance depression by 'pure algal diet'
CDWFRAC	0.4	[$gC.gDW^{-1}$]	Klepper (1989), mod.	Carbon fraction of dry weight mussel
CHLFBOT	1.1	[-]	Smaal & Haas (1997)	Factor to calculate CHLF near bottom from water value
CNRATMIN	4.7	[-]	Smaal & Vonck (1997)	Maximum C/N ratio in mussel body
DRYTOFLESH	4	[-]	Van Stralen (pers. com)	From dry weight to cooked flesh weight
FCCHL	0.03	[$gC.gCHLF^{-1}$]		Carbon to chlorophyll ratio
FLABDET	0.2	[-]		Fraction labile detritus in detPOM
GPTMAX	10	[h]	Bayne et al. 1989	Maximum gut passage time
GPTMIN	1.5	[h]	Bayne et al. 1989	Minimum gut passage time
NCELLCOL	300	[-]		Number of cell per Phaeocystis colony
NDWFRAC	0.1	[$gN.gDW^{-1}$]	Klepper (1989), mod.	Nitrogen fraction of dry weight mussel
NDWGAM	0.125	[$gN.g^{-1}$ DW]	Bayne (1976)	Nitrogen fraction in gametes mussel DW
OSWTODSW	20	[-]	Van Stralen (pers. com)	From organic shell weight to dry shell weight
PHDIAM	0.5	[mm]		Diameter of Phaeocystis colony
POCBOT	1.4	[-]	Smaal & Haas (1997)	Factor to calculate POC near bottom from water concentration
POCTOPOM	2.5	[$gDW.gC^{-1}$]		From POC to POM
SC	0.5	[h^{-1}]	Willows (1992), mod.	Digestive investment rate
SN	0.5	[h^{-1}]	Willows (1992), mod.	Digestive investment rate
SpSesMass	0.001	[$gSeston.mm^{-3}$]		Specific seston mass
TPMBOT	2	[-]	Smaal & Haas (1997)	Factor to calculate TPM near bottom from water value

Table G-8. List of calibrated parameters. All parameters are assumed to have a uniform distribution, of which the minimum and maximum values are given.

Parameter	Minimum	Maximum	Units	Reference	Comment
Acr	0.03	0.05	[$m^3.g^{-1}$ d^{-1}]	Smaal et al. (1997)	Allometric A-parameter clearance
AEmax	0.6	0.9	[-]	Bayne et al. (1989)	Absorption efficiency
Aexcr	0.0003	0.00045	[$gN.g^{-1}$ d^{-1}]	Smaal, Vonck (1997)	Allometric A-parameter excretion
Agc	12	20	mm^3 g^{-1}	Bayne et al. (1989)	Allometric A-parameter gut content
Ares	0.0028	0.0004	[$gC.g^{-1}$ d^{-1}]	Smaal et al. (1997)	Allometric A-parameter respiration
Ashell	25	35	[$mm.g^{-1}$DW]	Van Stralen (pers. com.)	Allometric A-parameter in Lshell to DSW
Bcr	0.4	0.6	[-]	Smaal et al. (1997)	Allometric B-parameter clearance
Bexcr	0.6	0.7	[-]	Smaal et al. (1997)	Allometric B-parameter excretion
Bgc	0.4	0.7	[-]	Bayne et al. (1989)	Allometric B-parameter gut content
Bres	0.5	0.9	[-]	Smaal et al. (1997)	Allometric B-parameter respiration

Bses	0.25	0.35	[-]	-	Parameter to calculate fPOMing from fPOMses
Bshell	0.3	0.5	[-]	-	Allometric B-parameter in Lshell to DSW
Dshell	0.000007	0.0007	[$gTWW.mm^{-3}$]	Van Stralen (pers. com.)	Parameter to calculate TWW from Lshell
GamGenTrig	0.15	0.5	[-]	Bayne (1984)	Minimum fraction glycogen to trigger spawning
GamRate	0.001	0.02	[d^{-1}]	-	Specific rate StorC to GamC per day
Kstor	3	50	[-]	Bayne & Newell (1983), mod.	Parameter in logistic function for repr. effort
KStorExp	0.001	0.005	[-]	Bayne & Newell (1983), mod.	Exponent in logistic function for repr. effort
Lcubic	2.5	3.5	[-]	Van Stralen (pers. com.)	Exponent to calculate TWW from Lshell
TempGam	8	12	[°C]	Hummel, Boogaards, 1989	Minimum temperature for spawning
MucusCoef	0.0001	0.01	[-]	-	Mucus production parameter (rejection)
OSWperSOM	0.1	0.3	[-]	-	Fraction of NetProdC to OSWC
Q10cr	1	1.5	[-]	Rueda et al. (2005)	Q10 parameter clearance
Q10res	1	1.5	[-]	Smaal et al. (1997)	Q10 parameter respiration
ResAbsC	0.02	0.07	[$gC.g^{-1}OIR$]	Hawkins & Bayne (1992)	Absorption related respiration rate parameter
StorCtresh	0	0.2	[-]	-	Fraction of storage not used for gametogenesis

G.5 Ontological layer 5: model knowledge for projects

G.5.1 Template

The knowledge of *ontological layer 4* has been used in several version of the simulation model for bivalve ecophysiology. The structural concepts presented in *ontological layer 5* are rather generic as they aim to provide vocabulary to describe a series of simulation models (and model versions) on bivalve ecophysiology and ecological processes. Table G-9 gives an overview of the structural concepts in *ontological layer 5*.

Table G-9. Ontology with *model knowledge for projects* (*ontological layer* 5). This template will be used here for *bivalve ecophysiological simulation models* (*ontological layer* 4). All terms in *italic* are defined components of other *ontological layer*s (0-3).

Concepts	Meaning
Model application name	Name and number of the *model application*.
Used in *object system*	Name of the *object system*.
Used for *problem*	Description of the main characteristics of the *problem*.
Model application purpose	Purpose of the *model version*.
Model application context	Context of the *model version*.
Model application references	References to journal papers, reports and other explicit scientific publications.
Knowledge *specializations*	Adaptations and/or specializations of the knowledge content in a less specialized *ontological layer*.

G.5.2 Instantiations for bivalve ecophysiological models used in projects

G.5.2.1 Simplified ecophysiology in SMOES

In the simulation model, SMOES (Simulation Model Oosterschelde EcoSystem) mussels and cockles are entities in an ecosystem (*object system*) with a substantial effect of that ecosystem. A part of the ecophysiological knowledge of bivalves as described in Table G-4, Table G-5, Table G-6, Table G-6, Table G-7 and Table G-8) has been gathered in a large ecosystem study in the Oosterschelde SW Netherlands (Klepper, 1989, Herman and Scholten, 1990, Scholten *et al.,* 1990, Van der Tol and Scholten, 1992, Klepper *et al.,* 1994, Scholten and Van der Tol, 1994, Van der Tol and Scholten, 1998). The project characteristics are described in Appendix F, Table F-5. The project characteristics are described in Table G-10, as instance of *ontological layer 5*.

Table G-10. Model ontology instantiation for the bivalve ecophysiological model elements in SMOES (*ontological layer* 5). All terms in *italic* are defined components of other *ontological layer*s (0-4). *Ontological layer* 5 consists of instances of ontological 4.

Concepts	Meaning
Model application name	SMOES (versions 1989-1994)
Used in object system	The estuarine ecosystem of the Oosterschelde, SW Netherlands, 1982-1987, described in Chapter 8, section 8.3.5.2.
Used for problem	Investigate the ecological effects of the building of a storm surge barrier in the mouth of the Oosterschelde

Model application purpose	SMOES aimed at simulating the main carbon and nutrients flows in the Oosterschelde ecosystem before the building of the storm surge barrier and to predict any ecological changes because of the storm surge barrier, based on the prediction of these main carbon and nutrient fluxes.
	
	SMOES distinguished 4 spatial compartments with transport between them.
	
	The total list of physical, biological and chemical processes included in the model is numerous and is described in Klepper (1989). A summary is given in (Klepper et al., 1994).
	
	SMOES included the following ecological *object system entities*:
	• *State variables* (all spatial compartments):
	    o *Diatoms* (i.e. silicate containing *phytoplankton*)
	    o Flagellates (i.e. *phytoplankton* that does not contain silicate)
	    o Benthic *diatoms*
	    o *Zooplankton*
	    o Dissolved silicate
	    o Particulate silicate
	    o Dissolved nitrogen
	    o Oxygen
	    o *Detritus*
	
	• *Time series* or functions, based on expert knowledge (all spatial compartments) for:
	    o Meiobenthos (function)
	    o Deposit feeders (function)
	    o Filterfeeders:
	        ▪ Biomass development of 2 year classes of mussels (function)
	        ▪ Standing stocks of mussel numbers in each year class and for each spatial compartment (time series)
	        ▪ Biomass development of 3 year classes of cockles (function)
	        ▪ Standing stocks of cockle numbers in each year class and for each spatial compartment (time series)

Model application context	Context was defined by two research projects: BALANS (1980-1988) and EOS (1988-1991). The two projects aimed at an analysis of the structure and functioning of the changing Oosterschelde estuary in perspective of the management of the water body after the building of the storm surge barrier in the mouth of the estuary (Nienhuis and Smaal, 1994). In the BALANS project the focus was on the pre-barrier situation, while EOS evaluated the post-barrier ecosystem.
Model application references	• Klepper and Scholten (1988) • Klepper (1989) • Klepper et al. (1991, 1994) • Herman and Scholten (1990) • Scholten *et al*. (1990) • Van der Tol and Scholten (1992, 1998) • Scholten and Van der Tol (1994)
Knowledge specializations	The model described in section G.4 is based on the description of the *ecophysiology* of *mussels* and *cockles* in SMOES, but in SMOES it was simpler including per year class: • Temperature and size dependent *clearance* • *Filtration* of food • *Pseudofaeces production* • *Absorption* (here called assimilation) • *Faeces production* • Temperature and size dependent *respiration* (this includes body size related *basal respiration* and activity related *routine respiration*) Based on these processes a *scope for growth* is calculated as: • *Scope for growth = assimilation – respiration* Besides the differences in the details of the *ecophysiological processes* between SMOES and the *mathematical model* of *bivalve ecophysiology* described in Chapter 9, section 9.3.4, the *bivalve ecophysiology* in SMOES is only a small (but essential) part of an *ecosystem model*. The *mathematical model* of *bivalve ecophysiology* described in Chapter 9, section 9.3.4 provides more details and can be plugged in a larger (*ecosystem* or carrying capacity model) to cope with the ecophysiology of *bivalves*. Furthermore, of the three types of *clearance depression*, only the one due to high *silt* content in *seston* is included.

G.5.2.2 EMMY: an ecophysiological model of Mytilus edulis *L.*

Three versions of the ecophysiological model EMMY (Ecophysiological Model of Mytilus edulis) have been used to investigate the ecophysiological response of bivalves to varying ecological conditions. The knowledge used to develop EMMY is discussed in Chapter 8. The mathematical model of EMMY is summarized in *ontological layer 4*, as instances of the concepts in *ontological layer 1, 2* and *3*.

According to the format provided by the template presented in section G.5.1, Table G-9 the characteristics of three versions of EMMY are described in Table G-11 (EMMY, version 1.6), Table G-12 (EMMY, version 2.0) and Table G-13 (EMMY, version 2.7), as instances of the concepts in Table G-9.

Table G-11. Model ontology instantiation for the EMMY model (*ontological layer* 5), as applied in Smaal and Scholten (1997). All terms in *italic* are defined components of other *ontological layer*s (0-4). *Ontological layer* 5 consists of instances of ontological 4.

Concepts	Meaning
Model application name	EMMY, version 1.6
Used in object system	A single mussel in the western part of the Oosterschelde, SW Netherlands, period 1982-1987.
Used for problem	Integrating quantitative, experimental and observation knowledge.
Model application purpose	Test the ecophysiological responses in EMMY that enable mussels to handle varying ecological conditions for the Oosterschelde ecosystem in the period 1982-1987, in terms of: • Food availability and quality • Temperature
Model application context	Context was defined by the research for Smaal (1997).
Model application references	• Smaal and Scholten (1997)
Knowledge specializations	Including the following 'extra' state variables compared to: • BloodC • BloodN • SpanwC • SpawnN

Table G-12. Model ontology instantiation for the EMMY model (*ontological layer* 5), as applied in Scholten and Smaal (1998). All terms in *italic* are defined components of other *ontological layer*s (0-4). *Ontological layer* 5 consists of instances of ontological 4.

Concepts	Meaning
Model application name	EMMY, version 2.0
Used in object system	A single mussel in three different ecosystems: • The western part of the Oosterschelde, SW Netherlands, 1982-1987; • The bay of Marennes-Oléron, France, 1983-1984; • Upper South Cove, Canada, 1991.
Used for problem	Testing the mechanisms, proposed by Scholten and Smaal (1998) in which mussels deal with varying food concentrations.
Model application purpose	Test the ecophysiological responses in EMMY that enable mussels to handle varying ecological conditions, as can be found in different ecosystems with substantial mussel culture and differing especially in food concentrations.
Model application context	Partially funded by EU Concerted Action AIR3-CT94-2219 'Trophic capacity of coastal zones for rearing oysters, mussels and cockles'. Also partially SEO funded by RIVO-DLO. This version of EMMY is the result of discussions on a NATO-workshop at RIKZ in the summer of 1997.
Model application references	• Scholten and Smaal (1998)
Knowledge specializations	Including the following 'extra' state variables: • BloodC • BloodN • SpanwC • SpawnN

Table G-13. Model ontology instantiation for the EMMY model (*ontological layer* 5), as applied in Scholten and Smaal (1999). All terms in *italic* are defined components of other *ontological layer*s (0-4). *Ontological layer* 5 consists of instances of ontological 4.

Concepts	Meaning
Model application name	EMMY, version 2.7
Used in object system	A single mussel in • The western part of the Oosterschelde, SW Netherlands, period 1982-1987. • 2 duplicates of low nutrient concentration seawater (North sea) and associated low mussel food concentrations (in mesocosms for 7 months) • 2 duplicates of medium nutrient concentration seawater (North sea) and associated medium mussel food concentrations (in mesocosms for 7 months) • 2 duplicates of high nutrient concentration seawater (North sea) and associated high mussel food concentrations (in mesocosms for 7 months) These scenarios reflected lower concentrations of nitrogen and phosphorous in the North Sea, as is described in Chapter 8, table 8-9.
Used for problem	Improving EMMY to deal with extreme values in food concentrations.
Model application purpose	This version aimed at: • Simplifying EMMY • Extending the range of food concentrations which can be handled by a mussel in EMMY
Model application context	SEO funding by RIVO-DLO.
Model application references	Scholten and Smaal (1999)
Knowledge specializations	This version of EMMY is similar to the model described in Table G-4, Table G-5, Table G-6, Table G-6, Table G-7 and Table G-8

G.5.2.3 COCO: an ecophysiological model of Cerastoderma edule *(L.)*

COCO, an ecophysiological model of *Cerastoderma edule* (L.) has been based on EMMY 2.0, which is a version without state variable 'blood', but with state variable SpawnC. All nitrogen related state variables, processes and other variables were left out, as being irrelevant. The cockles described by the model live on intertidal flats and can easily remove a surplus of nitrogen during low tides. An extra parameter has been added to control the effect of high and low tides by determining submersion as fraction of the day. The instance of COCO that has been used by Rueda et al. (2005) is summarized in Table G-14.

Table G-14. Model ontology instantiation for the COCO model (*ontological layer* 5), as applied in Rueda *et al.* (2005). All terms in *italic* are defined components of other *ontological layer*s (0-4). *Ontological layer* 5 consists of instances of ontological 4.

Concepts	Meaning
Model application name	COCO, version 1.1, which is based on EMMY 4.0
Used in object system	A single cockle in the western part of the Oosterschelde, SW Netherlands, period 1982-1987.
Used for problem	Integrating quantitative, experimental and observation knowledge.
Model application purpose	Test the ecophysiological responses in EMMY that enable mussels to handle varying ecological conditions in terms of: • Food availability and quality • Temperature
Model application context	The research project in which COCO has been developed was funded by a Marie-Curie training research grant of the European Commission, within the project SIMCERE (FAIR GT97-4525).
Model application references	• Rueda *et al.* (2005)

Knowledge specializations	COCO differs from EMMY 4.0 in many details, i.e. other values of parameters and a new parameter to express which part of the day cockles are submersed.

G.6 References

Bayne, B.L. and R.C. Newell, 1983. Physiological energetics of marine molluscs. The mollusca, vol 4: Physiology.

Bayne, B.L., 1976. Marine mussels - their ecology and physiology. International Biological Programme, 506 pp.

Bayne, B.L., 1984. Aspects of reproductive behaviour within species of bivalve molluscs. In: W. Engels, et al. (Eds.), Advances in invertebrate reproduction, Elsevier Science Publ., Amsterdam, pp. 357-366.

Bayne, B.L., A.J.S. Hawkins, E. Navarro and I.P. Iglesias, 1989. Effects of seston concentration on feeding, digestion and growth in the mussel Mytilus edulis. Mar. Ecol. Prog. Ser. 55, 47-54.

Cellier, F.E., 1991. Continuous System Modeling. Springer-Verlag, New York, ISBN 0387975020, 755 pp.

Edmonds, B., 2005. Simulation and Complexity - how they can relate. In: V. Feldmann, K. Mühlfeld (Eds.), Virtual Worlds of Precision - computer-based simulations in the sciences and social sciences, Methoden, Lit Verlag, Münster-Hamburg-Berlin-Wien-London, ISBN 3-8258-6773-0, pp. 5-32.

Hawkins, A.J.S. and B.L. Bayne, 1992. Physiological interrelations, and the regulation of production. In: E. Gosling (Ed.), The mussel Mytilus: ecology, physiology, genetics and culture, Developments in Aquaculture and Fisheries, Elsevier, Amsterdam, pp. 171-222.

Herman, P.M.J., Scholten, H., 1990. Can suspension-feeders stabilise estuarine ecosystems? In: Barnes, M., Gibson, R.N. (Eds.), Trophic relations in the marine environment. Aberdeen University Press, Aberdeen, pp. 104-116.

Hummel, H. and R.H. Boogaards, 1989, Changes in the reproductive cycle of the cockle Cerastoderma edule after disturbance by means of tidal manipulation. In: J.S. Ryland, P.A. Tyler (Eds.), Reproduction, Genetics and Distributions of Marine Organisms: 23rd European Marine Biology Symposium, Swansea, 5-9 September 1988, School of Biological Sciences, University of Wales, 133-136.

Klepper, O. and H. Scholten, 1988. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (SW Netherlands), DGW/DIHO, Middelburg, 278 pp.

Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary, DGW-Ministry of Transport and Public Works. Wageningen University, Wageningen, 270 pp.

Klepper, O., H. Scholten and J.P.G. Van de Kamer, 1991. Prediction uncertainty in an ecological model of the Oosterschelde estuary, S.W. Netherlands. Journal of Forecasting 10, 191-209.

Klepper, O., Van der Tol, M.W.M., Scholten, H., Herman, P.M.J., 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part I: description and uncertainty analysis. Hydrobiologia 282/283, 437-451.

Kramer, M.R. and H. Scholten, 2001, The Smart approach to modelling and simulation. In: A.W. Heemink, L. Dekker, H.d.S. Arons, I. Smit, T.L.v. Stijn (Eds.), PROCEEDINGS OF EUROSIM 2001, SHAPING FUTURE WITH SIMULATION, The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering., Delft, The Netherlands, June 26-29, 2001, TU Delft, 6 pages on CD-ROM, ISBN: 90-806441-1-0.

Minsky, M.L., 1965. Matter, mind, and models, Proceedings of International Federation of Information Processing Congress, vol. 1, Spartan Books, pp. 45-49.

NEN-6260-1, 2002. Termen en definities van modelleren, Nederlands Normalisatie Instituut, Delft, Normcommissie 390 147 19 'Watermodellering', werkgroep 390 147 1901 'basis principes modelleren', Design Norm NEN 6260-1, 19 pp.

Nienhuis, P.H. and A.C. Smaal (Eds.), 1994. The Oosterschelde estuary. Developments in Hydrobiology 97, H.J. Dumont, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-2817-5, Reprinted from Hydrobiologia, vols. 282/283, 597 pp.

Prins, T.C., A.C. Smaal and A.J. Pouwer, 1991. Selective ingestion of phytoplankton by the bivalves Mytilus edulis L. and Cerastoderma edule (L.). Hydrobiol. Bull. 25, 93-100.

Refsgaard, J.C. and H.J. Henriksen, 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27, 71-82.

Rueda, J.L., A.C. Smaal and H. Scholten, 2005. A growth model of the cockle (Cerastoderma edule L.) tested in the Oosterschelde estuary (The Netherlands). Journal of Sea Research 54, 276-298.
Russell, S. and P. Norvig, 1995. Artificial intelligence: a modern approach. Prentice Hall, Englewood Cliffs, NJ, 932 pp.
Scholten, H. and A.C. Smaal, 1998. Responses of *Mytilus edulis* L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.
Scholten, H. and A.C. Smaal, 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.
Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.
Scholten, H., O. Klepper, P.H. Nienhuis and M. Knoester, 1990. Oosterschelde estuary (S.W. Netherlands): a self-sustaining ecosystem? In: D.S. McLusky, V.N. De Jonge (Eds.), North Sea-estuaries interactions. Kluwer Academics Publishers, Dordrecht, pp. 201-215.
Smaal, A.C. and A.P.M.A. Vonck, 1997. Seasonal variation in C, N and P budgets and tissue composition of the mussel Mytilus edulis. Mar. Ecol. Prog. Ser. 153, 167-179.
Smaal, A.C. and F. Twisk, 1997. Filtration and absorption of Phaeocystis cf. globosa by the mussel Mytilus edulis L. J. Exp. Mar. Biol. Ecol. 33-46.
Smaal, A.C. and H. Scholten, 1997. EMMY: an ecophysiological model of *Mytilus edulis* L. In: A.C. Smaal (Ed.), Food supply and demand of bivalve suspension feeders in a tidal system. Rijksuniversiteit Groningen, Groningen, pp. 147-190.
Smaal, A.C. and H.A. Haas, 1997. Seston Dynamics and Food Availability on Mussel and Cockle Beds. Estuar. Coast. Shelf Sci. 45, 247-259.
Smaal, A.C., 1997. Food supply and demand of bivalve suspension feeders in a tidal system, Wiskunde en Natuurwetenschappen. Rijksuniversiteit Groningen, Groningen, 237 pp.
Smaal, A.C., A.P.M.A. Vonck and M. Bakker, 1997. Seasonal variation in physiological energetics of Mytilus edulis and Cerastoderma edule of different size calsses. J. Mar. Biol. Assoc. U.K. 77, 817-838.
Van der Tol, M.W.M. and H. Scholten, 1992. Response of the Eastern Scheldt ecosystem to a changing environment: functional or adaptive? Neth. J. Sea Res. 30, 175-190.
Van der Tol, M.W.M. and H. Scholten, 1998. A model analysis on the effect of decreasing nutrient loads on the biomass of benthic suspension feeders in the Oosterschelde (SW Netherlands). Journal of Aquatic Ecology 31, 395-408.
Willows, R.I., 1992. Optimal digestive investment: a model for filterfeeders experiencing variable diets. Limnol. Oceanogr. 37, 829-847.
Zeigler, B.P., 1976. Theory of modelling and simulation. Robert E. Krieger Publishing Company, Malabar, Florida, ISBN 0-89874-808-9, 435 pp.

Appendix H Testing the ontological framework

Content of Appendix H, belonging to chapter 10

H.1 Test results modelling ontology

The combination of the test types of section 10.4.2 (Table 10-1) have been used to evaluate the criteria of section 10.4.3 (Table 10-6) selected for each knowledge-based system component, being the *KB structure*, *KB process decomposition*, *KB content*, *MoST*, *Training material and Help* and *KB technology*. The results are presented in Table H-1.

Table H-1. Tested aspects of the *modelling ontology* with the results per criterion and per test type used.

<table>
<tr><th>#</th><th>Test criteria</th><th>Test types
Test results</th></tr>
<tr><td colspan="3">1 KB structure</td></tr>
<tr><td colspan="3">Not many comments on the ontological structure of the KB have been received. Ontologies are hard to understand for domain experts without knowledge engineering experience. Useful indications on how to change the ontology emerged indirectly when putting the pieces of modelling knowledge into instances of the ontology.</td></tr>
<tr><td>1a</td><td>Correctness</td><td>Project discussion:
No bugs detected.
Internal KB:
Not looked at.
Reviews:
Not looked at.
Scientific output:
Not commented by referees or by any reader. Some referees find it very interesting and promising.</td></tr>
<tr><td>1b</td><td>Completeness</td><td>Project discussion:
No major shortcomings detected and minor ones have been filled-in a next version.
Internal KB:
Small gaps found and subsequently filled.
Reviews:
Medium sized shortcomings in QA-evaluation ‘score-boards’ found, which have subsequently only partly removed.
Scientific output:
Not commented by referees or by any reader. Some referees find it very interesting and promising.</td></tr>
<tr><td>1c</td><td>Consistency</td><td>Project discussion:
No inconsistencies detected.
Internal KB:
Some inconsistencies have been detected and were subsequently repaired.
Reviews
No major inconsistencies detected.
Scientific output
Not commented by referees or by any reader. From scientific discussions there appeared an interest for automatic consistency checking.</td></tr>
<tr><td>1d</td><td>Granularity</td><td>Project discussion
Granularity level has been appreciated as adequately detailed.
Internal KB
In general the granularity level has been appreciated as adequate, although in some occasions it appeared difficult to decide whether some ‘action’ is a task or an activity.
Reviews
The reviewers did not comment the granularity level.
Scientific output</td></tr>
</table>

		The chosen granularity has been discussed briefly and no reviewer has commented it.
1e	Flexibility	Changing ontology *Changes in the ontological structure did require only a few hours work to change the structure (some tasks deleted, some tasks added, some activities changed into tasks and some tasks into activities).*
2 KB process decomposition		
*Several times we received requests for changing the structure of the modelling process. These requests came from project partners and from the 'wider modelling society'. All remarks were carefully evaluated and several have been used to improve the decomposition of the modelling process in **tasks** and of **tasks** into **activities**. These changes were mainly related to the order of the tasks, their dependencies, but also to the decomposition in tasks and the activities associated with the tasks. Implementing these changes was quite easy and not time consuming, because of the flexibility provided by the ontological approach. Typically, substantial changes required a few hours to a single day of work to incorporate the changes in MoST and its KB.*		
2a	Correctness	Project discussion *The correctness of the process decomposition has been discussed on many project meetings, leading to some changes in the decomposition and the associated KB. These were easy to implement because of the ontological approach. The resulting decomposition is therefore rather mature, but it will be always a topic for discussion in the process to reach consensus on it in the professional community.* Internal KB *Most of the changes in the modelling process decomposition were initiated by the partners that developed the content of the modelling KB. Their careful reading and evaluation of all (intermediate) versions of the KB made their opinion very constructive.* Reviews *Some comments of reviewers include the terminology of the decomposition, e.g.* parameter optimization *has been suggested, instead of* parameter estimation, *but such suggestions appeared to be arbitrary and strongly depending on someone's disciplinary background.*
2b	Completeness	Project discussion *Some of gaps in the decomposition have been detected in this way.* Internal KB *Most gaps were found with this test, although the initial decomposition still persist too a large extent probably due as the initial decomposition has been based on the Dutch GMP decomposition (Van Waveren et al., 1999). The identified gaps were mainly at the* activity level *and some at the* task level. Reviews *One reviewer[1] had some remarks on the completeness of the decomposition, based on his own experiences. Some of these have been included, others seemed to be irrelevant.*
2c	Redundancy	Project discussion *The project discussions resulted in intended redundancy in* methods*, as these can be use full for various* tasks *and for*

[1] This was Hugh Middlemis, as he has developed own (Australian) guidelines for groundwater modelling (Middlemis, 2000).

		various activities. Internal KB *Some redundancies have been detected and removed, most at the* activity level *of the decomposition.* Reviews *No redundancies have been detected here.*
2d	Consistency	Project discussion *Here no inconsistencies appeared.* Internal KB *Some inconsistencies have been detected and repaired, but the internal testers mentioned that it is not possible to find all inconsistencies and stated that identifying and removing inconsistencies is a continuous process, unless it can be automated.* Reviews *Some minor inconsistencies (at the* activity level*) have been found and removed.*
2e	Transparency	Project discussion *The project discussions concluded that the decomposition is transparent, especially in the flowchart view in MoST. The* step *and* task *level were perceived as transparent and capturing the intuitions of the experts. At the* activity level*, the decomposition is less directly transparent because of the details.* Internal KB *No remarkable results have to be mentioned here.* Reviews *The reviews resulted in a very positive conclusion on transparency of the decomposition.*
2f	Granularity	Project discussion *The granularity has been object of discussion during the whole HarmoniQuA project. The decomposition in itself with steps, tasks and activities as the main decomposition, allows to access and use the decomposition at several layers with increasing details. The final decomposition is perceived as detailed enough to be not too abstract.* Internal KB *During the project more details were added in order too avoid an abstract decomposition.* Reviews *One reviewer wanted more details and the other two perceived the decomposition as sufficient detailed.*
3 KB content		
Feedbacks on the content of the decomposition elements included long lists of errors, wishes and comments. But all respondents so far appreciated the guidance provided by the KB and found it useful, especially for novice users of MoST.		
3a	Correctness	Internal KB *A long series of smaller and bigger errors have been found and corrected. The setup of the KB content has not been changed much in four years.* Case studies *Only minor changes have been proposed based on the experiences in the 21 case studies. A major comment comprised the idea that a modelling project should be redefined (which tasks are relevant in the present project). This interacts with the basic idea that the guidelines should*

		act as template for the modelling process, which can be fine-tuned at the beginning of a project. Reviews *A long series of suggestions to improve the content of the KB originated from one reviewer, mainly concerning auditing and task/activities for auditors. Some of these have been implemented and others were included in the overall product (KB, MoST, training material) as spreadsheets that can be coupled to the model journal of a modelling team.* Workshops *Some suggestion arose from workshops. These have been carefully examined and some have been implemented partly (mainly on calibration).* User questionnaires *The respondents evaluated the correctness of the KB content with a 7.8 (scale 1-10).* Scientific output *In several papers a part of the content of the KB has been included and that part has been accepted by the referees. The KB is too comprehensive to include completely.*
3b	Completeness	Internal KB *Two new tasks have been added in a rather late stage of the HarmoniQuA project. The first task (set-up scenarios) was required, as in many modelling projects evaluating scenarios and determining the best is a major issue. The second task (pos-audit) mainly consists of determining whether or not a post-audit has to be done when new data is available or after implementing the results of a model study.* Case studies *The case studies generate a long series of comments of which some in regard to the completeness. The overall perception found the KB guidance very useful and complete.* Reviews *The reviewers found a relative small number of gaps, most related to validation and to auditing modelling projects.* Workshops *Some gaps have been found during the workshops, but most appeared to be no gaps. Missing 'tasks' were included at a lower decomposition level as 'activities'.* User questionnaires *The respondents evaluated the completeness of the KB content indirectly by evaluating the amount of the guidance with a mark of with a 6.8 (scale 1-10). This rather mediocre result can be explained from the comments of the respondents. Many indicated that the guidance was too comprehensive and detailed.* Scientific output *Only parts of the KB content on model-based water management have been published. These parts have been appreciated very well, but this does not evaluate the completeness of the KB content.*
3c	Redundancy	Internal KB *Some remarks have been made here on the methods. Some methods can be used in various* tasks/activities *and these should be defined only once. An example is the* method *'sensitivity analysis', which is used in the* tasks *'Uncertainty*

		Analysis of Simulation', 'Uncertainty Analysis of Calibration and Validation', 'Select Calibration Parameters'. Case studies *Some minor redundancies have been detected in the case studies and subsequently removed.* Reviews *Several redundancies have been detected by the reviewers and subsequently removed.* Workshops *No redundancies have appeared from the workshops.* User questionnaires *An indication on how users evaluated redundancy is given by the mark for the amount of guidance and by some additional comments (just as used for completeness). The respondents gave a mark of 6.8 (scale 1-10) and perceived that many details are duplicate at the* activity level. Scientific output *No redundancies have been found in the process of reviewing and publishing scientific output.*
3d	Consistency	Internal KB *Several inconsistencies have been identified.* Case studies *Several inconsistencies have been identified.* Reviews *No inconsistencies have been identified.* Workshops *No inconsistencies have been identified.* Scientific output *No inconsistencies have been found in the process of reviewing and publishing scientific output.*
3e	Meaningfulness	Internal KB *The partners involved in this internal test were professional modelers and they found some small shortcomings related to the meaningfulness of the KB content. A large part of their work was related to this topic, as they improved the content to the expertise of its intended users.* Case studies *A few deficiencies showed up here and could easily be repaired.* Reviews *Each of the three reviewers had his own disciplinary background and their remarks on this issue were closely related to that (one is modeler in groundwater quantity and quality and is interested most in evaluating and auditing; one is sociologist with a specific interest in participatory process (information, consultation and co-decision making); the third has a background at the interface of legal responsibilities and guidelines with special interest in the legal consequences of guidelines and model documentation on validation issues). Some of their remarks have been implemented and others were neglected to balance the KB content for its intended users.* Workshops *The workshops showed that the meaningfulness of the content of the KB is acceptable for most of the intended users, but this result may be skewed by the fact that workshop*

		participants had a more positive attitude to the KB content, due to the single fact that they decide to participate in a workshop. User questionnaires *The respondents evaluated the meaningfulness of the KB content indirectly by evaluating if the guidance is understandable. Their mark for this issue was 6.0 (scale 1-10), which indicates that the guideline is rather unclear. This result is mainly due to student respondents, unfamiliar with modeling.* Scientific output *No remarks on the meaningfulness of the KB content came from reviewers of the scientific output.*
3f	Necessity	Professional survey *On opinion on MoST and KB (n=221):* • *38 % vital* • *66 % helpful* • *67 % interesting* • *16 % tiresome* • *45 % laborious* • *5 % useless*
3g	Acceptance	Internal MoST *In the process to get agreement on the KB a clear acceptance of all partners in HarmoniQuA have been achieved.* Reviews *Although comments of the reviewers were received, the overall KB content was acceptable for the reviewers, but is was also clear that they have not been involved in the KB development and so not participated in the process to come to a compromise to get the agreement.* Case study *Many (smaller) issues came up from the case studies and these have been used to improve the KB content* Workshops *The workshops resulted in a series of smaller comments, which have been partly used. The overall opinion of workshop participants was positive (but this may probably skewed by the fact that participating coincides with a positive attitude).*
4 MoST		
The results for criterion 'Adequately supporting daily practice of professionals' accumulated in the first test series were promising and directed the redesign of MoST to a more powerful level. These tests led to a long series of small suggestions that have been discussed and partially implemented. The first test series also identified more important shortcomings. The modelling support provided by MoST was insufficient in two aspects. The version used for these tests was too much focused on single users and on monodisciplinary projects. Extra functionality for modelling teams and for multidisciplinary projects has been implemented in the full version of MoST. The second test series with the full version of MoST provided other needs for change. Applying MoST and its KB in university courses resulted in a similar request, i.e. for multi-user support and multidisciplinary application domains. *Many testers (professionals) and students wanted MoST to enable them to work at a higher level: not only fulfilling tasks by doing activities, but also detailing what team members do at a task level only. In the latter case the activities are just headings in the model journal contribution for that task.*		
4a	Correctness	Internal MoST *Many smaller errors have been detected in this way and all have been solved so far. Especially to let MoST work the first time after installation required inventiveness of the user.*

		These problems have been solved adequately in newer versions. Case studies *Many smaller errors have been detected in this way and all have been solved so far.* Reviews *Several errors have been detected and all have been solved so far.* Workshops *Several errors have been detected and all have been solved so far.* Courses *No errors have been detected in this way.* User questionnaires *The respondents evaluated the correctness of MoST on each of the following aspects (marks on a scale of 1-10):* *1. Model project initialization 7.5* *2. Providing guidance 7.8* *3. Glossary 8.1* *4. Monitoring general 7.9* *a. Tasks 7.7* *b. Activities 7.6* *c. Reporting:* *i. Selection 7.3* *ii. Generating 7.3* *iii. Viewing/printing 7.8*
4b	Reliability	Internal MoST *In the earlier versions of the client-server version of MoST there were some reliability problems, which have been solved, except for installation on non-windows PCs.* Case studies *Most case studies worked with local projects (single-user) and here no reliability problems occurred.* Reviews *The reviewers worked with local projects (single-user) and here no reliability problems occurred.* Workshops *The workshops worked with local projects (single-user) and here no reliability problems occurred.* Courses *Here severe problems showed up with the first client-server version of MoST in online projects. Most had to do with the intensity of the traffic on the server, which was too slow to handle server requests within acceptable performance levels. This caused unforeseen user response (repetitive saving projects with unwanted server behaviour.*
4c	Functionality	Internal MoST *Many requests for additionally functionality have been proposed by this test type. Almost all have been realised in the later versions.* Case studies *The case studies resulted in the most requiring demands for extended functionality, of which the most important are:* • ***Multi-user options**. These have been realized by a client-server approach in which each user runs a client-application on a pc and the model journal is*

		stored on a project server and accessible for and shared all project team members. The project server application is responsible for synchronizing the model journal changes of each team member in the single and shared model journal on the project server. • ***Multi-domain options****. Users felt the need for functionality that allows handling projects in more than one domain. In case of more than one domain, these domains can be handled tightly coupled (in time) or having their own subproject speed.* Reviews *From the reviews some suggestions for additional functionality emerged. The most important one (extra spreadsheet based functionality to add extra (detailed) score boards, has been realised.* Workshops *From this test type as series of extra functionalities has been, of which a* help-system *is the most noteworthy. This has been realised. A second important wanted feature was the possibility to work at two levels of detailing, being at* task-level *and at* activity-level *during project execution. This has also been implemented.* Courses *Students usually find MoST too complex and they appreciate the training (especially the screen-recordings). They also prefer to work at the task-level in project execution. This is probably related to the short projects they were involved and the lack of modelling skills and expertise.* User questionnaires *The respondents did not compare MoST with its requirements, but compare MoST with their expectations, leading to the following marks (on a scale of 1-10):* *1. Model project initialization 7.3* *2. Providing guidance quality 6.5* *3. Glossary 7.3* *4. Monitoring general 7.1* *a. Tasks 6.9* *b. Activities 6.9* *c. Reporting:* *i. Selection 7.1* *ii. Generating 7.0* *iii. Viewing/printing 6.9*
4d	Adequacy	Case studies *Case studies in which teams of more than one person had to do a modelling project appreciated MoST, as it fitted in their daily practice and supported the exchange of information between team members.* Reviews *One reviewer was very positive, while the other two looked from different angles (legal consequences and participatory processes). The latter two had initially reservations, but later (after the project ended) more enthusiastic on MoST's practical use.*
4e	Learnability	Courses *Students did not experienced MoST (and its KB on*

<table>
<tr><td></td><td></td><td>modelling) as useful for their course work. Oppositely, their teachers experienced it as very helpful to learn students modelling, as it acted as a framework in training novices how to model.</td></tr>
<tr><td>4f</td><td>Necessity</td><td>Professional survey[2]
On opinion on MoST and KB (n=221):
• 38 % vital
• 66 % helpful
• 67 % interesting
• 16 % tiresome
• 45 % laborious
• 5 % useless
Stakeholder survey
On importance of model documentation such as provided by MoST's model journals: most respondents stated that detailed information of how the model was applied was useful, important or crucial:
• 26 % Crucial
• 42 % Important
• 30 % Useful
• 2 % Not needed</td></tr>
<tr><td>4g</td><td>Acceptance</td><td>Reviews
The reviewers had the strong opinion that potential users have to take a hurdle, which works as threshold. Once taken they will be enthusiastic, otherwise hesitant to adopt the approach providing by MoST and its modelling KB.
Case study
The incentive of extra revenues when using MoST and its KB in a case study appeared to be helpful and persuasive in the attitude to use MoST and its KB in other (professional) modelling projects.
Workshops
The workshop attendants had and expressed a similar opinion: the complexity of the tool will hinder acceptance. Most participants expected that using MoST and its KB will be laborious and an overhead to their work. After the workshop they still thought it is helpful and beneficial, both in terms of the quality of the modelling work and the ease of running (complex) model projects.</td></tr>
<tr><td colspan="3">5 Training material + Help[3]</td></tr>
<tr><td colspan="3">Testing the training material led to many relatively small changes. Many of the national workshops for professionals are scheduled for this year, so answers on criterion 'usefulness professional' are not available yet. But using the training material in student courses ('usefulness students') showed very promising results. Students learn very quickly (a few hours) how to use MoST in a training case study and apply it in their problem oriented education projects, in which small groups of students have to solve environmental problems with a model. The guidance provided by the KB directed them effectively through the network of tasks, of which modelling projects usually consist. This approach also proved to be more efficient than the textbook approach on Good Modelling Practices used in the same courses in the past.</td></tr>
<tr><td>5a</td><td>Correctness</td><td>Workshops
Only small errors appeared from the workshops, which could easily be corrected.</td></tr>
</table>

[2] This is equal to evaluating the criterion *necessity* of the KB (content). The professional response questionnaire did not ask for the separate attitudes of the respondents on the KB and MoST. Therefore the results have to be seen as the attitude to both.

[3] The training material and the help-system were not yet available at the time of the reviews.

		Courses *Only small errors appeared from the courses, which could easily be corrected.* User questionnaires *The respondents evaluated the correctness of the training material on each of the following aspects (marks on a scale of 1-10):* *1. Website works without errors 8.3* *2. Screen recordings work without errors 8.1* *3. Presentations are shown without errors 8.3* *4. The additional information is provided without errors 8.0*
5b	Usefulness professionals	Workshops *There are three programs to train professionals at workshops, each with their own usefulness to train professionals in the use of MoST and its KB:* • Demonstration *(2h): this type appeared to be a good instrument to show the potential of MoST and its KB, but not enough to let professionals really use it in their daily practice. The latter would require that they did the short workshop training in a self-study context.* • Short workshop *(4-6h): This training program appeared to be adequate in teaching professionals how to use MoST and its KB, except for working in a multi-user and multi-domain setting. The latter would require additional training as is provided in the long workshop training setting.* • Long workshop *(12-16h): this type is not used so far and therefore no conclusions on its usefulness can be drawn.* User questionnaires *The respondents evaluated the usefulness of the training material on each of the following aspects (marks on a scale of 1-10):* *1. Website functions as I want and expect 7.5* *2. Website is easy to understand 7.5* *3. Website is helpful 7.6* *4. Screen recordings provide information as I want and expect 7.6* *5. Screen recordings are easy to understand 8.0* *6. Screen recordings are helpful 7.5* *7. Presentations provide information as I want and expect 7.4* *8. Presentations are easy to understand 7.7* *9. Presentations are helpful 7.4* *10. The additional information is as I want and expect 7.4* *11. Getting the additional information is easy 7.6* *12. The additional information is helpful 7.5*
5c	Usefulness students	Courses *The training program used for students so far, is a variant to the* short workshop program *adapted for modelling courses at Wageningen University (see Table 10-4). Students work for 4 hours to become familiar with MoST and its modelling KB. Teachers considered this training as very successful, as*

		it is an efficient and effective way to let novices become familiar with complex software that deals with rather abstract issues. User questionnaires *See 5b, as for this issue no differentiation has been made between professionals and students.*
6 KB technology		
Using the HarmoniQuA technology in AquaStress showed that technology related criteria 'Is the KB ontological structure reusable?' and 'Is the tool MoST useful for other processes?' are fulfilled so far, although successfully using the technology in AquaStress and other processes in future will enhance confidence in reusability.		
6a	Reusability KB structure	Reuse *The ontological structure of the KB (process – step –task – activity, with many more concepts) appeared to be reusable in the EC-financed project AquaStress. It is used in several test cases at the test sites* Velt and Vecht *(Netherlands) and* Iskar *(Bulgaria). Obviously, the KB content is replaced by knowledge on water stress mitigation (the topic of all AquaStress research).*
6b	Reusability MoST	Reuse *MoST is used without any adaptation so far and its application is without problems. Despite this ease of use several extensions of MoST's functionality are required by the AquaStress users, including:* • *More flexible handling of dedication aspects: in MoST several dedication aspects are used:* o *User type* o *Domain* o *Job complexity (not used anymore)* o *Application purpose (not used)* *This structure is too rigid and other processes (e.g. water stress mitigation in AquaStress) require different dedication aspects.* • *Extra feature to let MoST support planning and managing projects too. This requires small extra ontological concepts, e.g. deadlines and some extra functionality, e.g. progress compare to deadlines.*

H.2 Testing the problem and object system ontology

Two aspects of the *problem and object system ontology* have been tested: the *structure*, i.e. how the various concepts are related, and the *content*, i.e. the semantics of concepts. These aspects of the *problem and object system ontology* have been tested with 3 types of tests, which are summarized in chapter 10 (Table 10-8).

The criteria used to evaluate the *model ontology* aspects are similar to those used to test the modelling ontology (see chapter 8, Table 10-6). The test results are summarized in Table H-2.

Table H-2. Tested aspects of the *problem and object system ontology* with the results per criterion and per test type used.

#	*Test criteria*	*Test results*
1 Problem and object system ontology structure		
Testing the problem and object system ontology structure *was difficult, as the best form of testing is using the ontology in applications and by tools. The few tests performed for this ontology did not reveal many shortcomings and errors. Despite the modest level of testing, the development of such an ontology has value in itself.*		
1a	Correctness	Face evaluation:

		Especially the correctness Bugs detected in this way have been corrected. Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology without revealing errors.* Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
1b	Completeness	Face evaluation: *Deficiencies detected concern mainly branches (for other domains and paradigm) not filed in so far. Other deficiencies have been corrected.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology without revealing shortcomings.* Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
1c	Redundancy	Face evaluation: *Redundancies detected in this way have been corrected.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology without revealing redundancies.* Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
1d	Consistency	Face evaluation: *Inconsistencies detected by* face evaluation *have been corrected.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology without revealing inconsistencies.* Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
1e	Granularity[4]	Face evaluation: *Detected flaws in the granularity by* face evaluation *have been improved.* Application: *The granularity at the lowest level can best been determined by applying the ontological concepts at the lowest level (most decomposed concepts) by using its instances. This has been done in series of models on bivalve ecophysiology without revealing an improper granularity.* Reuse: *So far the problem and object system ontology is not reused (i.e. filled with factual knowledge on other domains).*
2 Problem and object system ontology content		

[4] A well chosen granularity facilitates a direct use of concepts and their instances in a model.

The problem and object system ontology *was applied to develop a series of models on bivalve ecophysiology. From applying the* content *of the* problem and object system ontology, *it can be concluded that the knowledge organized in this way is useful to develop simulation models. Therefore this leaf of the ontological framework can be assumed as validated, although in a very limited sense.*		
2a	Correctness	Face evaluation: *Correcting the content of the* problem and object system *ontology is not so difficult. Errors in the factual knowledge on bivalve ecophysiology have been detected and corrected.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology without revealing errors.* Reuse: *So far the problem and object system ontology is not reused (i.e. filled with factual knowledge on other domains).*
2b	Consistency	Face evaluation: *Inconsistencies are harder to detect by* face evaluation. *The inconsistencies found in this way, have been corrected.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology and this revealed some inconsistencies, which were repaired.* Reuse: *So far the problem* and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
2c	Acceptance	Face evaluation: Face evaluation *is not a proper test type to investigate acceptance of the* problem and object system ontology content *by a substantial part of the expert community. Indirectly* face evaluation *tests consistency between published expert knowledge and ontology content. If one assumes that published knowledge is shared by a substantial part of the expert community, the content of the knowledge base will also be accepted.* Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology. Publication of these models in (refereed) journals facilitates indirectly* acceptance *the content of the* problem and object system ontology. Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*
2d	Meaningfulness	Face evaluation: *Experts doing the* face evaluation *understood the content of the* problem and object system ontology *and its content can therefore be regarded as* meaningful. Application: *The* problem and object system ontology *is applied to develop a series of models on bivalve ecophysiology. This can be seen as an indirect test on* meaningfulness. Reuse: *So far the* problem and object system ontology *is not reused (i.e. filled with factual knowledge on other domains).*

H.3 Testing the model ontology

Two aspects of the model ontology have been tested: the *structure*, i.e. how the various concepts are related, and the *content*, i.e. the semantics of concepts. These aspects of the model ontology have been tested with 3 types of tests, summarized in chapter 10 (Table 10-8).

The criteria used to evaluate the *model ontology* aspects are similar to those used to test the modelling ontology (see chapter 8, Table 10-6). The test results are summarized in Table H-3.

Table H-3. Tested aspects of the *model ontology* with the results per criterion and per test type used.

#	Test criteria	Test results
1 Model ontology structure		
Testing the model ontology structure *was difficult, as the best form of testing is applying an ontology in applications and by tools that use the ontology. Despite the modest level of testing, the development of such an ontology has value in itself.*		
1a	Correctness	Face evaluation: *Bugs detected in this way have been corrected.* Application: *There was a close connection between developing the model ontology and developing models based on it. In this way the ontology reflects what models actually nee. The models obviously use only a part of the ontology and therefore the rest of the ontology is not tested in this way.* Publishing: *No errors have been detected by publishing models based on the ontology.*
1b	Completeness	Face evaluation: *Deficiencies detected in this way have been corrected.* Application: *Deficiencies detected when using the ontology in developing models have been included.* Publishing: *No deficiencies have been detected by publishing models based on the ontology.*
1c	Redundancy	Face evaluation: *Redundancies detected in this way have been corrected.* Application: *No redundancies have detected when using the ontology in developing models have been included.* Publishing: *No redundancies have been detected by publishing models based on the ontology.*
1d	Consistency	Face evaluation: *Inconsistencies detected by* face evaluation *have been corrected.* Application: *Inconsistencies detected by* face evaluation *have been corrected.* Publishing: *No inconsistencies have been detected by publishing models based on the ontology.*
1e	Granularity[5]	Face evaluation: *Detected flaws in the granularity by* face evaluation *have been improved.* Application:

[5] A well chosen granularity facilitates a direct use of concepts and their instances in a model.

		The granularity at the lowest level can best been determined by applying the ontological concepts at the lowest level (most decomposed concepts) by using its instances in a model. Publishing: *Publications of models based on the* model ontology *did not reveal flaws in the granularity.*
2 Model ontology content		
Testing the model ontology content *was also difficult, as no tools have been developed that use the model ontology content. As soon as such tools have been developed more shortcomings will probably be detected.*		
2a	Correctness	Face evaluation: *Bugs detected by* face evaluation *have been corrected.* Application: *The part of the* model ontology content *of which instances were used in developing a series of bivalve ecophysiology models is tested in this way and errors found in this way have been corrected.* Publishing: *No errors have been detected by publishing models based on the ontology..*
2b	Consistency	Face evaluation: *Inconsistencies detected by* face evaluation *have been corrected.* Application: *Inconsistencies detected by* applying *the* model ontology content *for other have been corrected.* Publishing: *No inconsistencies have been detected by publishing models based on the ontology*.
2c	Acceptance	Face evaluation: *Acceptance of the* model ontology content *by a substantial part of the modelling community is partly tested by face evaluation and acceptance by the experts doing the face evaluation contributes to a wider acceptance.* Application: Applying the *content* of the *model ontology* (*ontological layer 3*) does not test on acceptance by the professional community of experts in bivalve ecophysiology. Publishing: *Publishing facilitates members of the wider modelling community to comment of the* model ontology content *and in this way helps its testing. No comment could be used to improve the* model ontology content.
2d	Meaningfulness	Face evaluation: *The intended users should understand the content of the model ontology, which includes the experts doing the face evaluation.* Application: *So far, I am the only person that has applied the* model ontology. *Therefore, applying the* model ontology *has hardly improved the* meaningfulness. Publishing: *Publishing did not revealed problems with the meaningfuness of the* model ontology..

H.4 References

Middlemis, H., 2000. Murray-Darling Basin Commission. Groundwater flow modelling guideline. In. South Perth. Western Australia: Aquaterra Consulting Pty Ltd. www.harmoniqua.org/public/Reports/existing%20guidelines/MBDC_Groundwater_Modeling.pdf

Van Waveren, R.H., S. Groot, H. Scholten, F. Van Geer, H. Wösten, R. Koeze and J. Noort, 1999. Vloeiend modelleren in het waterbeheer. Handboek Good Modelling Practice. STOWA/RWS-RIZA, Utrecht/Lelystad, ISBN 90-5773-056-1, 149 pp.

Appendix I User questionnaire results

Content of Appendix I, belonging to chapter 10

I.1 General

	n[1]	R[2]	%	C[3]
n =	50			
Type of training:				
Students:		44		
Demonstration course		4		
Short course		21		
Long course		19		
Professionals:		5		
Demonstration course		0		
Short course		2		
Long course		3		
Platform:				
Windows98		3	6%	
Windows2000		9	18%	
WindowsNP		1	2%	
WindowsXP		39	78%	
Other Windows		0	0%	
Linux		0	0%	
HP-UX		0	0%	
Solaris		0	0%	
Other Unix		0	0%	
Mac OS X		0	0%	
Year of experience (av)	44	4.52		
Type of user:				
Student		40	80%	
Professional modeller		4	8%	
Interested scientist		6	12%	
Software engineer		2	4%	
Knowledge engineer		0	0%	
Teacher / trainer		3	6%	
I work (also) in the HarmoniQuA project		1	2%	
I am interested in learning how to use MoST because:				
I think MoST will improve my modelling.		9	18%	
It is a part of the course I am following.		35	70%	
I want to find out if MoST is useful for me.		8	16%	
I want to investigate the scientific soundness of MoST's methodology		3	6%	
Other		3	6%	
number of comments				5

[1] n = number of respondents.
[2] R = result.
[3] C = number of comments.

I.2 Overall impression

I.2.1 Impression training material

	n	R	%	C
Training web material is useful introduction to MoST.		42	84%	31
The training web material requires a HarmoniQuA trainer		23	46%	28
Training was of sufficient duration		39	78%	23

I.2.2 Impression MoST

	n	R	%	C
I would like to use MoST in my modelling work.		24	48%	38
I believe the guidance it offers will be useful.		38	76%	29
I believe the monitoring functionality it contains will be useful.		31	62%	26
I believe the reporting functionality it contains will be useful.		43	86%	29
I believe MoST will enhance the quality of modelling work.		37	74%	34

I.3 Detailed answers

I.3.1 Details training material

I.3.1.1 Website

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Website works without errors.	42	8.3		
Website functions as I want and expect.	42	7.5		
Website is easy to understand.	42	7.5		
Website is helpful.	42	7.6		
What changes should be made to the training website?				18

I.3.1.2 Screen recordings

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Screen recordings work without errors.	39	8.1		
Screen recordings provide information as I want and expect.	39	7.6		
Screen recordings are easy to understand.	39	8.0		
Screen recordings are helpful.	39	7.5		
What changes should be made to the screen recordings?				18

I.3.1.3 Presentations

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Presentations are shown without errors.	41	8.3		
Presentations provide information as I want and expect.	41	7.4		
Presentations are easy to understand.	41	7.7		
Presentations are helpful.	41	7.4		
What changes should be made to the presentations?				21

I.3.1.4 Additional information on the training website

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
The additional information is provided without errors.	34	8.0		
The additional information is as I want and expect.	34	7.4		
Getting the additional information is easy.	34	7.6		
The additional information is helpful	34	7.5		
What changes should be made to the additional information?				18

I.3.2 Details MoST

I.3.2.1 User Interface (UI)

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
UI works without errors.	45	6.7		
UI functions as I want and expect.	45	7.0		
UI is easy to understand.	46	7.0		
What changes should be made?				23

I.3.2.2 Model Project Initialization (MPI)

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
MPI works without errors.	37	7.5		
MPI functions as I want and expect.	37	7.3		
MPI is easy to understand	37	7.2		
What changes should be made?				14

I.3.2.3 Guidance

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
The modelling guidance is provided without errors.	40	7.8		
The amount of guidance provided by MoST is as I want and expect.	39	6.8		
The quality of the guidance provided MoST is as I want and expect.	39	6.5		
The guidance is understandable.	40	6.0		
What changes should be made?				23

I.3.2.4 Glossary

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
The glossary works without errors.	36	8.1		
The glossary functions as I want and expect.	36	7.3		
The glossary is easy to use.	35	7.6		
The glossary is helpful.	36	6.8		
What changes should be made?				22

I.3.2.5 Monitoring the modelling tasks and activities

I.3.2.5.1 General

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
The monitoring component works without errors.	35	7.9		
The monitoring component functions as I want and expect.	35	7.1		
The monitoring component is easy to understand.	34	7.4		
What changes should be made?				19

I.3.2.5.2 Tasks

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Performing a task works without errors.	41	7.7		
Performing a task functions as I want and expect.	41	6.9		
How to perform a task is easy to understand.	41	6.4		
What changes should be made?				22

I.3.2.5.3 Activities

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Performing an activity works without errors.	38	7.6		
Performing an activity functions as I want and expect.	38	6.9		
How to perform an activity is easy to understand.	38	6.3		
What changes should be made?				18

I.3.2.5.4 Other activities/task related issues

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
The methods provided are useful.	0	-		
The methods provided are as I want and expect.	36	6.5		
The methods provided are easy to understand.	1	7.0		
What changes should be made to the methods?				-
The sensitivities and pitfalls are useful.	27	6.9		
The sensitivities and pitfalls are easy to understand	27	6.6		
What changes should be made to the sensitivities and pitfalls?				13
The other information is useful	25	7.0		
The other information is easy to understand	25	7.0		
What changes should be made to the other information?				10

I.3.2.5.5 Reporting

1 = I do not agree at all, 10 = I fully agree

	n	R	%	C
Selecting what to include in a report works without errors.	32	7.3		
Selecting what to include in a report functions as I want and expect.	32	7.1		
How to select what include in a report is easy to understand.	32	7.1		
What changes should be made to selecting what to include in a report?				12

Generating reports works without errors.	29	7.3	
Generating reports functions as I want and expect.	29	7.0	
Generating reports works is easy to understand.	29	7.1	
What changes should be made to generating reports?			11
Viewing and printing reports works without errors.	28	7.8	
Viewing and printing reports functions as I want and expect.	28	6.9	
How to view and print reports is easy to understand.	28	7.6	
What changes should be made to viewing and printing what to include in a report?			10

I.4 Conclusion

	n	R	%	C
If you have additional comments, please fill these in here:				22
Please inform me on the results of the enquiry.		11	22%	
I do not object to answer additional questions related to MoST.		10	20%	
Please treat my information as confidential.		9	18%	

Summary

Managers and policy makers in our society are confronted with decision making problems of great complexity. This complexity is caused by a variety of requirements of various stakeholders of the problems, by the problem solving opportunities and by the inherent properties and scope of the problem situation. Mastering this complexity in an effective and efficient manner nowadays often calls for mathematical models and knowledge based systems that are able to adequately support the decision or policy making processes.

The first step of model-based problem solving projects is that managers have to arrive at a shared vision on the nature and extent of a modelling project (here mainly focused on continuous simulation models), in which solutions have to be found for a management problem stated *a priori*. Such a vision entails the scope of the study, the solution approach, expected results, duration, costs and resources used. Thereafter, for a commissioned project, the problem is to execute it in compliance with its specifications including Quality Assurance (QA) issues. Transparency should be guaranteed and projects should be easier to audit and reconstruct.

Increased quality assurance awareness and requirements for modelling projects are caused and fuelled by a multitude of problems and bad experiences with model based studies in the past. There are several reasons for these problems. These include ambiguous terminology, a lack of mutual understanding between key-players, bad practice in regard to input data, inadequate model set-up, insufficient calibration/validation, model use outside of its scope, insufficient knowledge on some object system processes, miscommunication between the modeler and the end-user, overselling of model capabilities, confusion on how to use model results in the decision process and a lack of documentation and transparency of the modelling process itself.

An additional complicating factor in the problem context is associated with the increased need for the support of multidisciplinary problem solving. Over the last decades changes in the character of model-based problem solving projects have been observed from monodisciplinary, single person and academic oriented research model studies into multidisciplinary, decision support oriented projects, in which teams consisting of members with different backgrounds and different roles that have to collaborate to complete the complex job in response to societal and academic requirements. Such projects may typically integrate several subprojects, belonging to various application domains and are often carried out by international, distributed teams. In addition, since a few years the complexity of model-based decision support has been extended even further and at present many studies have to include socio-economic impact assessments too. Moreover, there are often legal prerequisites to decision making that require public participation in the decision making process, e.g. the Water Framework Directive and similar legislation.

At present, modelling to support decision making should be seen as a process, in which multidisciplinary modelling teams work in multiple application domains, use methodology from various modelling paradigms and follow a participatory approach. It enables exploring more complex questions and addresses quality assurance requirements, but it also makes modelling more difficult. Team members with different scientific backgrounds (or even without specific expertise) encounter increasing communication problems. This makes managing multidisciplinary model-based (water) management projects a cumbersome affair.

Most initiatives to overcome problems related with modelling incidents in the Netherlands and in several other countries, have led to an increased interest in model related uncertainty, but there are many other lessons learned resulting in a variety of other approaches. The modelling problems that have been reported about can be summarized as follows:

- Ambiguous terminology and a lack of mutual and shared understanding between key-players (modelers, clients, auditors, stakeholders and concerned members of the public);
- Bad practice (careless handling of input data, inadequate model set-up, insufficient calibration and validation and model use outside of its scope);
- Lack of data or poor quality of available data;
- Insufficient knowledge about processes hindering ecological (biota) modelling;
- Miscommunication of the modeler to the end-user on the possibilities and limitations of the modelling project and overselling of model capabilities;
- Confusion on how to use model results in the decision process;
- Lack of documentation and transparency of the modelling process, leading to projects, which hardly can be audited or reconstructed;
- Insufficient consideration of economic, institutional and political issues and a lack of integrated modelling.

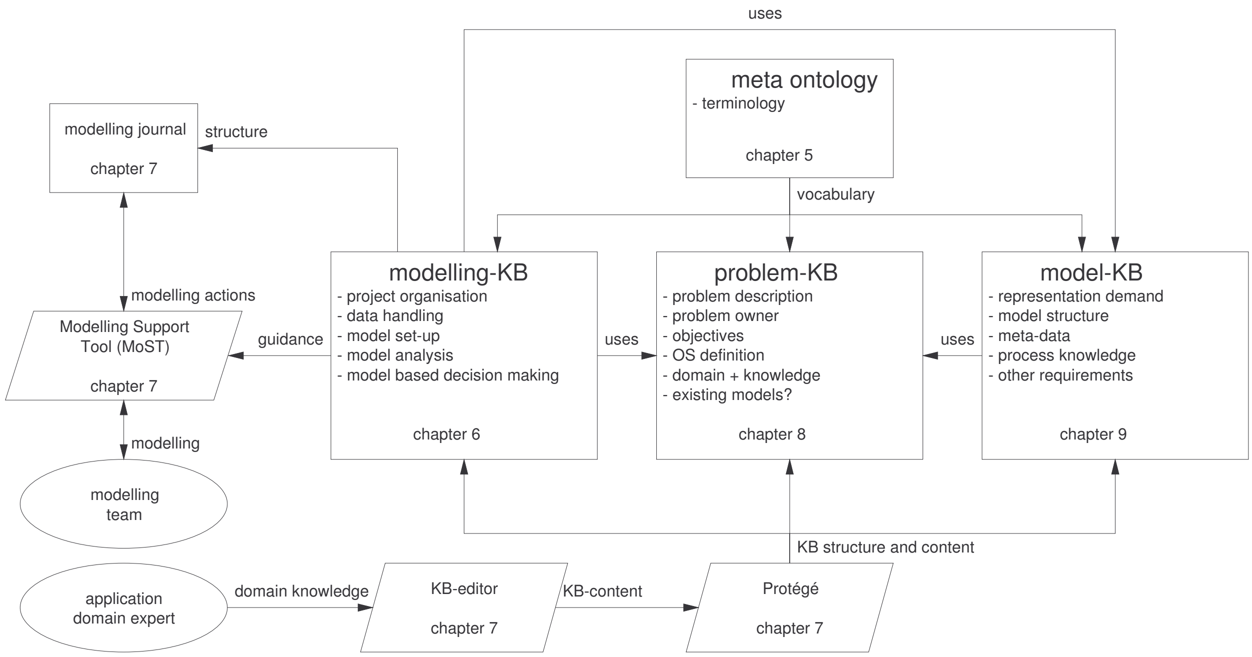
Reviews of the responses of the research community to these problems show recommendations providing scientific and technical guidance on how to decide on what to model and how to carry out various steps in the modelling work to achieve the best and most reliable results. Existing modelling guidelines, mostly nationally based, focus on a single domain in contrast to integrated models. Furthermore, these guidelines vary worldwide. Resulting model outcomes and decisions based on them are often non-transparent, irreproducible, non-auditable and not fully comparable among different countries.

Two other problems can be added to the above list:
- In the acquisition phase the problem owner has to select the modeler, i.e. a consultant or other organization that has the right expertise for the problem to be solved. In this tendering procedure the problem owner can make a completely wrong choice for a problem solver. This can cause various problems, including inadequate expertise to solve the problem.
- In the project start-up major selections (which problem related processes to include and which type of models to use) can put the problem-solving process on a completely wrong track.

The mentioned variety of modelling problems raised the basic question: *How to improve the quality of modelling in order to increase its credibility*. Most of the outlined problems deal with the *process dimension* of multidisciplinary modelling, but a consistent, well-structured view is also needed on the problem to be solved and on the models, which are instrumental in model-based problem solving. Therefore the basic question is further refined in Chapter 1 into three research questions aiming at describing the three dimensions (*problem* and its associated *object system*, *model* and *modelling*) as formal as possible to improve the credibility of models and modelling. These formal descriptions are the knowledge part of a framework that further consists of a meta-ontology with basic terminology and a tool box to set up and edit the ontological knowledge bases and to support organization, execution and management of modelling projects.

The research discussed here is design oriented. The remainder of this book can be divided in three parts. The first part (Chapters 2, 3 and 4) describes methodological issues. Chapters 5, 6, 7, 8 and 9 are the core part of the framework to support multidisciplinary teams in model-based problem-solving. The third part (Chapters 10 and 11) discusses the testing the framework and a general discussion.

Chapter 2 summarizes a series of scientific building blocks, including *system science*, *mathematical models*, *quality assurance*, *structured modelling and simulation*, *relational databases*, *knowledge-based systems and artificial intelligence*, *process defining technologies* and finally *software engineering and process modelling support tools*. These building blocks have been used to set up the framework. In Chapter 3 a major methodological basis the framework is introduced i.e. the knowledge engineering approach that uses ontologies to structure knowledge. Some terms are essential in setting up ontologies, e.g. *concepts* (everything that needs to be and can be described) and *relations* (how concepts are linked, e.g. concept B is a part of concept A). In Chapter 4 a monodisciplinary view on Kuhn's scientific paradigm is used to outline a similar multidisciplinary view, which is instrumental to set up the framework to support multidisciplinary teams in model-based problem-solving, described in the core part of this book.



Outline of the proposed framework to improve modelling. Rectangles represent ontological knowledge bases, parallelograms represent (software) tools and ovals represent (groups of) persons.

Chapter 5 introduces a *meta-ontology* that provides terminology for bootstrapping other ontologies.

Chapter 6 describes in detail how processes can be structured in a *process ontology*, which can be filled with pieces of knowledge. This is illustrated by filling the ontological process knowledge structure with (continuous simulation) modelling knowledge, especially from water management within the context of the HarmoniQuA project[1]. The process knowledge ontology is organized in layers of increasing specialization with the aim to achieve that the more basic, generic layers can be reused for other purposes and also that the more generic layers can be instrumental to discuss and communicate knowledge from the more detailed and specialized layers within the multidisciplinary team, consisting of members that are not specialists in that field. The ontology editor and knowledge base tool, Protégé, has been used to set up the structured ontologies. To help domain experts adding their expertise to the ontological knowledge bases in Protégé a web based Knowledge Base Editor has been developed that act as a front-end for Protégé.

Chapter 7 introduces and discusses the *Modelling Support Tool* (MoST) that supports the work of modelling teams in their daily practice by presenting modelling guidance from the modelling knowledge base, by monitoring what is done by each team member and by facilitating project management with detailed data on what is done by who in the project and on resources spent.

Chapters 8 and 9 describe a *problem ontology* and a *model ontology*, which have a similar layered structure of increasing specialization as the process ontology of Chapter 6. The *problem ontology* (Chapter 8) and its knowledge base are filled with knowledge on bivalve ecophysiology, i.e. how

[1] HarmoniQuA has been partially funded by the 5th Framework Programme of the European Commission.

the physiology of bivalves (e.g. mussels, cockles) responds to dynamic ecological conditions (e.g. food availability, temperature). The *model ontology* (Chapter 9) consists again of a layered structure of increasing specialization. The generic layers contain concepts of mathematical models in general and the more specialized layers contain concepts of model types of specific modelling paradigms, here continuous simulation models, instantiated for bivalve ecophysiology models.

Chapter 10 describes the testing of the framework to support multidisciplinary teams in model-based problem-solving. Although all elements are tested, emphasis is on the ontological modelling knowledge base and MoST, as these are the more mature parts of the framework.

Chapter 11 discusses the framework as a whole. It is concluded that a substantial part of the problems outlined in Chapter 1 have been solved. The basic question (*How to improve the quality of modelling in order to increase its credibility*) is answered by a design and partial implementation of a framework to support multidisciplinary teams in model-based problem-solving. The resulting framework can be seen as an answer to the research questions. This framework should not be seen as *the answer*, but as *an answer* to the needs of model-based problem solving projects and their (multidisciplinary) teams at hand, restricted to a specific setting and for a specific purpose. Another team in another project will have other wishes, another purpose and will require another instance of the framework. This book claims that at least a part of the ontological framework (if necessary filled with new content) is reusable.

Samenvatting

Managers en beleidsmakers worden geconfronteerd met besluitvormingsproblemen van grote complexiteit. Deze complexiteit wordt veroorzaakt door een scala aan eisen van allerlei belanghebbenden, door de diversiteit aan mogelijkheden om de problemen op te lossen en door de eigenschappen en de draagwijdte die horen bij de probleemsituatie. Om deze complexiteit op een effectieve en efficiënte wijze te beheersen zijn wiskundige modellen en op kennisgebaseerde systemen nodig die adequaat het besluit- of beleidsvormingsproces kunnen ondersteunen.

De eerste stap in projecten die modellen gebruiken om problemen op te lossen bestaat uit het verkrijgen van een gedeelde visie op het karakter en de omvang van een modelleerproject (hier is vooral gefocust op continue simulatiemodellen), waarin oplossingen gevonden moeten worden voor een van te voren geformuleerd managementprobleem. Een dergelijke visie omvat de draagwijdte van de studie, de benadering om tot een oplossing te komen, de te verwachten resultaten, tijdsduur, kosten en benodigde inzet. Vervolgens moet het (als het om een uitbesteed project gaat) worden uitgevoerd in overeenstemming met de specificaties waarbij tevens de kwaliteit geborgd moet worden. Transparantie moet worden gegarandeerd en projecten moeten gemakkelijk gecontroleerd en gereconstrueerd kunnen worden.

Een toegenomen kwaliteitsbewustzijn en de toegenomen eisen aan modelleringprojecten worden gevoed door problemen en slechte ervaringen met modelstudies in het verleden. Er zijn verschillende oorzaken voor deze problemen aan te wijzen, waaronder dubbelzinnige terminologie, wederzijds onbegrip tussen de hoofdrolspelers, onjuist omgaan met (meet)gegevens, waarnemingen), verkeerd opstellen van het model, onvoldoende kalibratie en validatie, modelgebruik buiten het toepassingsgebied, onvoldoende kennis over sommige processen van het te modelleren systeem, miscommunicatie tussen modelleur en de eindgebruiker, te hoog opgeven over de mogelijkheden van het model, onduidelijkheid over hoe modelresultaten te gebruiken in besluitvormingprocessen en een gebrekkig gedocumenteerd en niet-transparant modelleerproces.

Een extra complicerende factor hangt samen met de toegenomen behoefte aan het ondersteunen van het oplossen van multidisciplinaire problemen. Gedurende de laatste tientallen jaren kan men veranderingen waarnemen in het karakter van op modellen gebaseerde projecten om problemen aan te pakken, van monodisciplinaire, eenpersoons en academische, onderzoeksgerichte modelstudies, naar multidisciplinaire, besluitvormingondersteunde projecten, waarin teams van experts met verschillende achtergrond en een verschillende rol moeten samenwerken om de complexe klus te klaren in antwoord op vragen uit de maatschappij en de wetenschap. In zulke projecten komen deelprojecten samen die behoren tot verschillende toepassingsdomeinen en die worden uitgevoerd door teams uit verschillende organisaties en vaak uit verschillende landen. Sinds enkele jaren is de complexiteit van op modellen gebaseerde besluitvormingsondersteuning verder toegenomen en moeten ook vaak socio-economische gevolgen worden meegenomen. Bovendien zijn er vaak wettelijke voorschriften voor besluitvorming, die voorschrijven dat ook belanghebbenden betrokken moeten worden in het besluitvormingsproces, zoals de Europese Kaderrichtlijn Water en vergelijkbare wetgevingen.

Tegenwoordig kan modelleren beter worden gezien als een proces, waarin modelleerteams werken in diverse toepassingsgebieden tegelijk, methoden en technieken gebruiken afkomstig uit verschillende modelleerparadigmata en betrokkenen participeren in het proces. Op deze manier kunnen complexere vragen worden onderzocht zonder dat kwaliteitsborgingeisen uit het oog worden verloren. Het modelleren wordt wel ingewikkelder maar ook beter. Teamleden met een verschillende disciplinaire achtergrond (of zelfs zonder specifieke expertise) ondervinden meer

communicatieproblemen. Dit maakt het managen van multidisciplinaire, op modellen gebaseerde waterbeheerprojecten vaak een moeizame aangelegenheid.

De meeste initiatieven om de problemen te lijf te gaan die samenhangen met modelleerincidenten in Nederland en in verschillende andere landen hebben geleid tot een toegenomen belangstelling voor onzekerheid in modellen, maar daarnaast hebben andere benaderingen ook resultaat geboekt. De modelleerproblemen kunnen als volgt worden samengevat:

- Dubbelzinnige terminologie en een gebrek aan wederzijds en gedeeld begrip tussen de hoofdrolspelers (modelleurs, opdrachtgevers, auditoren, belanghebbenden en geïnteresseerden);
- Slecht modelleren (onzorgvuldig omgaan met invoergegevens, inadequaat opzetten van het model, onvoldoende kalibratie en validatie, het gebruik van het model buiten het toepassingsgebied);
- Te weinig gegevens beschikbaar of data van slechte kwaliteit;
- Onvoldoende kennis van de werkelijke processen maakt het modelleren van ecologie moeilijk;
- Miscommunicatie van de modelleur naar de eindgebruiker over de mogelijkheden en beperkingen van het modelleerproject en te hoog opgeven over de mogelijkheden van het model;
- Onduidelijkheid over hoe modelresultaten te gebruiken in besluitvorming;
- Gebrekkige documentatie en transparantie van het modelleringproces, dat daardoor moeilijk is te beoordelen en te reconstrueren;
- Onvoldoende rekening houden met economische, institutionele en politieke kwesties en geen geïntegreerd modelleren.

Overzichtsartikelen over de reactie van de onderzoeksgemeenschap op deze modelleerproblemen bevelen aan om wetenschappelijke en technische richtlijnen te verstrekken over hoe te besluiten over wat te modelleren en hoe alle verschillende stappen te nemen in het modelleerwerk om het beste en meest betrouwbare resultaat te verkrijgen. Bestaande modelleringrichtlijnen, meestal nationale, concentreren zich op één toepassingsdomein in plaats van op geïntegreerde modellen. Verder zijn deze richtlijnen overal anders. De verkregen modelresultaten en de genomen besluiten zijn vaak niet transparant, ze zijn niet reproduceerbaar, niet te beoordelen en tussen landen niet volledig met elkaar te vergelijken.

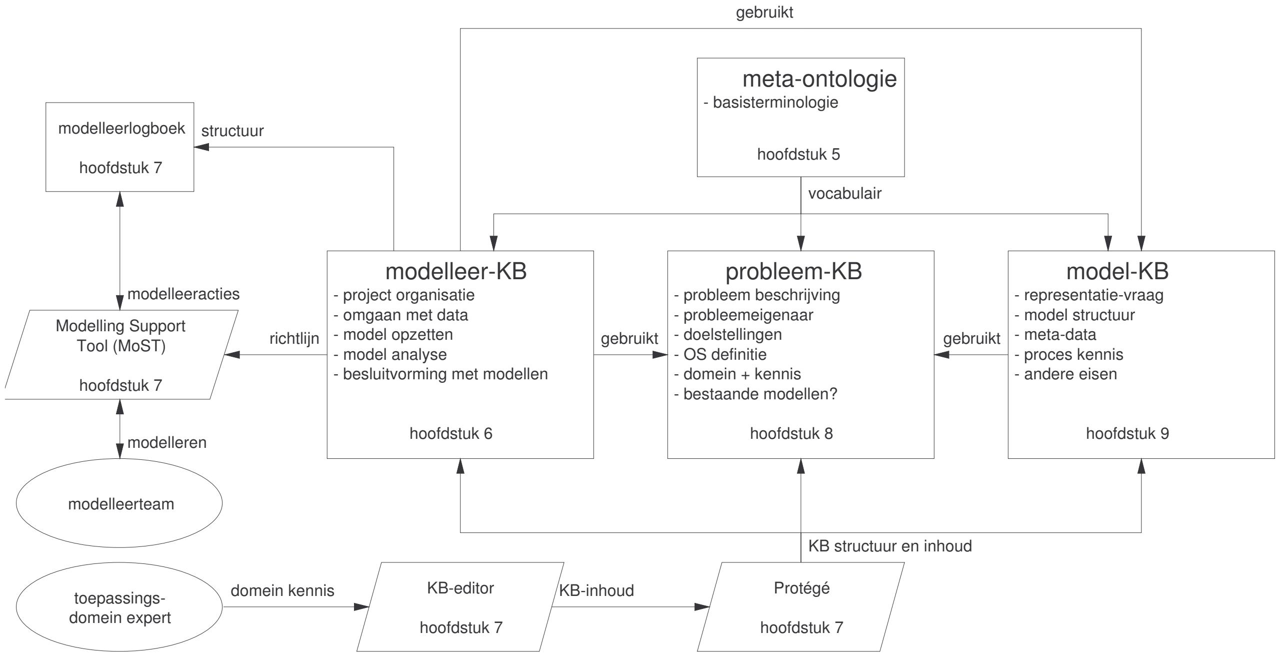
Twee andere problemen moeten worden toegevoegd aan bovenstaande lijst:

- In de acquisitiefase moet de probleemeigenaar een modelleur kiezen, dat wil zeggen een adviesbureau of andere organisatie die de juiste expertise heeft voor het probleem dat moet worden opgelost. In de aanbesteding kan de probleemeigenaar een volledig verkeerde keuze maken voor een organisatie om het probleem op te lossen. Dit kan allerlei moeilijkheden veroorzaken zoals onvoldoende expertise om het probleem op te lossen.
- In de beginfase van een project kunnen belangrijke keuzes (welke aan het probleem gerelateerde processen mee te nemen en welk type modellen te gebruiken) het probleemoplossend proces volledig op het verkeerde spoor zetten.

Dit scala aan modelleerproblemen roept de volgende basisvraag op: *Hoe de kwaliteit te verbeteren van modelleren om zo de geloofwaardigheid ervan te verhogen*. De meeste van de hier geschetste moeilijkheden gaan over de procesdimensie van multidisciplinair modelleren, maar ook het op te lossen probleem en het daarvoor te gebruiken model vereisen een consistente en gestructureerde visie. Daarom wordt in hoofdstuk 1 de basisvraag gespecificeerd in drie onderzoeksvragen, die beogen om de drie dimensies (*probleem* en het daarmee samenhangende object system, het *model* en het *modelleren*) zo formeel mogelijk te beschrijven om zo de geloofwaardigheid van modellen en modelleren te verhogen. Deze formele beschrijvingen vormen het kennisdeel van een raamwerk dat verder bestaat uit een meta-ontologie met basisterminologie en een gereedschapskist met software om een toepasselijke kennisbank op te zetten en te onderhouden en software om het organiseren, uitvoeren en beheer van modelleerprojecten te ondersteunen.

Het onderzoek is ontwerpgericht. De rest van het boek kan worden verdeeld in drie delen. Het eerste deel (hoofdstukken 2, 3 en 4) beschrijft de methodologie. De hoofdstukken 5, 6, 7, 8 en 9 vormen het hart van het raamwerk om multidisciplinaire teams te ondersteunen bij het oplossen van problemen met behulp van modellen. Het derde deel (hoofdstukken 10 en 11) bestaat uit het testen van het raamwerk en een algemene discussie.

Hoofdstuk 2 geeft een overzicht van de wetenschappelijke bouwstenen van het raamwerk om multidisciplinaire teams te ondersteunen bij het oplossen van problemen met behulp van modellen. De volgende bouwstenen zijn gebruikt: *systeemkunde*, *wiskundige modellen*, *kwaliteitsborging*, *gestructureerd modelleren en simuleren*, *relationele databases*, *op kennisgebaseerde systemen en kunstmatige intelligentie*, *technologieën om processen te definiëren* en tenslotte *software engineering en software om processen zoals modelleren te ondersteunen*. In hoofdstuk 3 wordt een belangrijke methodologische basis van het raamwerk geïntroduceerd, namelijk *kennisengineering* dat gebruikt maakt van ontologieën om kennis te structureren. Sommige begrippen zijn essentieel bij het ontwikkelen van ontologieën, namelijk *concepten* (alles dat moet en kan worden beschreven) en *relaties* (hoe concepten samenhangen, bijvoorbeeld concept B is een deel van concept A). In hoofdstuk 4 wordt een monodisciplinaire visie op Kuhn's wetenschappelijke paradigma gebruikt om een vergelijkbare multidisciplinaire visie te schetsen, die instrumenteel is bij het ontwikkelen van een raamwerk om multidisciplinaire teams te ondersteunen bij het oplossen van problemen met behulp van modellen, zoals beschreven is in het centrale deel van dit boek.



Schets van het voorgestelde raamwerk om modelleren te verbeteren. Rechthoeken geven ontologische kennisbanken weer, parallellogrammen geven (software) tools weer en ovalen geven (groepen van) personen weer.

Hoofdstuk 5 introduceert een meta-ontologie met de termen die nodig zijn om andere ontologieën te kunnen ontwikkelen.

Hoofdstuk 6 beschrijft in detail hoe processen kunnen worden gestructureerd in een *procesontologie*. Dit wordt geïllustreerd door de ontologische structuur voor proceskennis te vullen met kennis over (continue simulatie)modellen, vooral afkomstig uit het waterbeheer in de context van het HarmoniQuA project[1]. De ontologie met proceskennis is georganiseerd in steeds verder gespecialiseerde lagen, met als doel de meest basale, generieke lagen te kunnen gebruiken voor andere toepassingen. Verder zijn deze generieke lagen instrumenteel bij het bediscussiëren en communiceren van meer gespecialiseerde en gedetailleerde kennis binnen het multidisciplinaire team dat deels bestaat uit leden die geen specialist zijn op dat gebied. De ontologie en kennisbank ontwikkelomgeving, Protégé, is gebruikt om de gestructureerde ontologieën te ontwikkelen. Om domeinexperts te helpen om hun expertise toe te voegen aan de kennisbanken, is een *Knowledge Base Editor* ontwikkeld die het experts gemakkelijker maakt om de ontologische kennisbanken in Protégé te vullen.

Hoofdstuk 7 introduceert en bediscussieert de *Modelling Support Tool* (MoST) die het dagelijkse werk ondersteunt van modelleerteams door modelleerrichtlijnen aan te bieden, afkomstig uit de modelleerkennisbank, door bij te houden wat is gedaan door elk teamlid en door het management van dit soort projecten te ondersteunen met gedetailleerde gegevens over wat er door wie in het project is gedaan en hoeveel tijd en geld daarvoor is gebruikt.

De hoofdstukken 8 en 9 beschrijven een *probleemontologie* en een *modelontologie* die net als de procesontologie uit hoofdstuk 6 een gelaagde structuur hebben met steeds verder gespecialiseerde lagen. De *probleemontologie* (hoofdstuk 8) en de daarbij behorende kennisbank is gevuld met kennis over de ecofysiologie van tweekleppigen, d.w.z. hoe de fysiologie van tweekleppigen (zoals mosselen en kokkels) reageert op veranderende ecologische omstandigheden (zoals voedselbeschikbaarheid en temperatuur). De *modelontologie* (hoofdstuk 9) heeft ook een gelaagde structuur met steeds gespecialiseerdere lagen. De generieke lagen bevatten concepten met betrekking tot wiskundige modellen in het algemeen en de gespecialiseerdere lagen bevatten concepten behorend bij soorten modellen die tot specifieke modelparadigmata behoren, hier continue simulatiemodellen waarbij ecofysiologische modellen van tweekleppigen als voorbeeld zijn uitgewerkt.

Hoofdstuk 10 beschrijft het testen van het raamwerk om multidisciplinaire teams te ondersteunen bij het oplossen van problemen met behulp van modellen. Alle componenten zijn getest, maar de nadruk ligt op de ontologische kennisbank over modelleren en op MoST, omdat dit de meer voldragen delen zijn van het raamwerk.

Hoofdstuk 11 bediscussieert het raamwerk als geheel. De conclusie is gerechtvaardigd dat een belangrijk deel van de in hoofdstuk 1 geschetste problemen zijn opgelost. De basisvraag (*Hoe de kwaliteit te verbeteren van modelleren om zo de geloofwaardigheid ervan te verhogen*) is beantwoord door het realiseren van een ontwerp en een gedeeltelijke implementatie van het raamwerk om multidisciplinaire teams te ondersteunen bij het oplossen van problemen met behulp van modellen. Het resulterende raamwerk kan worden gezien als een antwoord op de onderzoeksvragen. Het raamwerk moet niet worden gezien als *het antwoord*, maar als *een antwoord* op de behoeften van projecten die problemen oplossen met behulp van modellen en de multidisciplinaire teams die daarin werken en dan ook nog binnen een bepaalde situatie en voor een bepaald doel. Andere teams en andere projecten hebben andere wensen, een ander doel en hebben behoefte aan een andere invulling van het raamwerk. Dit boek claimt dat minstens een deel van het raamwerk (zonodig gevuld met nieuwe inhoud) herbruikbaar is.

[1] HarmoniQuA is voor een groot deel gefinancierd binnen het 5^{e} kaderprogramma van de Europese Commissie.

‘Reeling and Writhing, of course, to begin with,’ the Mock Turtle replied;
'and then the different branches of arithmetic --
Ambition, Distraction, Uglification, and Derision.’
(Lewis Carroll, pseudonym of Charles Lutwidge Dodgson, mathematician and logician, 1832-1898)

Post Scriptum

Met een netje ongerechtigheden scheppend uit het zwembad van een huis in de Languedoc onder de mediterrane zon, heb ik hoofdstuk 8 geschreven. Dit hoofdstuk gaat deels over hoe tweekleppige schelpdieren hun voedsel zeven uit het water. Mijn schepnetje is minder efficiënt dan de kieuwen van een mossel, maar dat geeft meteen de tijd om na te denken. Filteren, sorteren en inbouwen in structuren. Mosselen kunnen dat gedachteloos, maar ik niet. Het schrijven van een boek vereist concentratie en veel gezwoeg en het is niet bij voorbaat duidelijk of het ergens toe leidt. Voor je het weet is het resultaat van dat gezwoeg nutteloos, behalve voor zijn auteur. Mosselen malen daar niet om.

Om op mijn leeftijd nog een promotie te willen afronden vereist meer concentratie dan de gemiddelde ADHDer op kan brengen. Dit wordt ruimschoots gecompenseerd door snelheid en energie van diezelfde ADHDer. Die energie is verdwenen (of om het wat positiever te zeggen: geïnvesteerd) in diverse, zeer uiteenlopende grote projecten:

- BALANS (M€ 5[1] om de effecten van de stormvloedkering op het ecosysteem van de Oosterschelde te voorspellen. Een deel van de inspanningen resulteerden in SMOES een ecosysteemmodel voor de Oosterschelde, samen met vooral Olivier Klepper (toen *DIHO*, later *RWS-DGW*), Aad Smaal (toen *RWS-DGW*, later *WUR-RIVO*, nu Wageningen Imares) en vele anderen).
- EOS (M€5 om de voorspellingen van BALANS te controleren, samen met vooral Marcel van der Tol (*RWS-RIKZ*) en Aad Smaal (toen *RWS-DGW*, later *WUR-RIVO*, nu Wageningen Imares).
- ECOLMOD: k€ 350[2] om samen met Bert-Jan de Hoop (toen *DIHO*, nu *WU*) en Peter Herman (*DIHO* dat nu *NIO-CEMO* heet) SENECA te ontwikkelen, een simulatie omgeving voor ecologische toepassingen met veel toeters en bellen voor gevoeligheidsanalyse, calibratie en onzekerheidsanalyse). Daarnaast nog een kleine bijdrage aan MOSES, een ecosysteemmodel voor het estuarium van de Schelde.
- SMART: k€ 175 een simulatieomgeving speciaal voor gebruik in het universitair onderwijs samen met vooral Mark Kramer, maar ook met anderen, waaronder Alexander Udink ten Cate (toen *WU* en *DLO*, daarna *Open Universiteit*, nu *Christelijke Hogeschool Windesheim*).
- HarmoniQuA[3]: M€ 2.5 om de kwaliteit te waarborgen van het gebruik van modellen ten behoeve van waterbeheer, met vooral Ayalew Kassahun (*WU*), Jens Christian Refsgaard (*GEUS*, Denemarken), George Zompanakis (*NTUA*, Griekenland), Theodor Kargas (*NTUA*, Griekenland), Gareth Old (*CEH*, UK), Johan Meerkerk (*Nympaea Support*) en Adrie Beulens (*WU*).
- AquaStress[4]: M€ 14 om *water stress* problemen op te lossen vooral rond de Middellandse Zee, samen met Ayalew Kassahun (*WU*), Michiel Blind (*RWS-RIZA*, nu *Deltares*), Peter Gijsbers (*WL|Delft Hydraulics*, nu *Deltares*), Henk Wolters (*RWS-RIZA*, nu *Deltares*), Nils Ferrand (*Cemagref*), Arno Krause (*WUR-Alterra*) en vele anderen.

In een groot aantal kleinere projecten kwamen diverse onderwerpen aan bod, waaronder (maar niet beperkt tot):

- Mijn vakantieproject in de jaren negentig over calibratie en onzekerheidsanalyse dat gebruikt maakt van de kennis van experts, samen met Marcel van der Tol (*RWS-RIKZ*)
- Een methode om de tochtigheid van koeien te voorspellen met behulp van vage logica samen met Wim Eradus (toen *ATO*) en Alexander Udink ten Cate (toen *WU* en *DLO*, daarna *Open Universiteit*, nu *Christelijke Hogeschool Windesheim*) als instrument om in te bouwen in melkrobots (gemiddeld winst per melkveehouder € 2500 per jaar)
- Kwaliteitsborging van modellen was aanvankelijk het thema van het onderzoek voor een proefschrift, vooral samen met Alexander Udink ten Cate (toen *WU* en *DLO*, daarna *Open Universiteit*, nu *Christelijke Hogeschool Windesheim*)

[1] Afgerond in miljoenen euro’s.
[2] Afgerond in duizenden euro’s.
[3] Deels gefinancierd door de Europese Commissie in het Vijfde Kaderprogramma.
[4] Deels gefinancierd door de Europese Commissie in het Zesde Kaderprogramma.

- GMP: k€ 90 om het Handboek *Good Modelling Practice* te schrijven, vooral samen met Harold van Waveren (*RIZA*), Frans van Geer (*TNO-NITG*), Simon Groot (*WL|Delft Hydraulics*) en Henk Wösten (*WUR-WUR-Alterra*)
- NEN-normen: k€ 40 voor Nederlandse normen ten behoeve van het modelleren in het waterbeheer, samen dezelfde mensen als het Handboek GMP.
- Het ontwikkelen van modellen voor de ecofysiologie van tweekleppige schelpdieren zoals mosselen en kokkels, samen met vooral Aad Smaal (toen *WUR-RIVO*, nu Wageningen Imares), maar ook José Rueda (Universidad de Málaga). Dit project gebeurde weliswaar vrijwel met gesloten beurs (een kleine bijdrage van het EU-TROPHEE-project en een kleine bijdrage van SEO gelden van het toenmalige *WUR-RIVO*), maar leverde toch een viertal publicaties op, waarvan drie in goede tijdschriften. Eén hiervan werd onderscheiden met de Dresscher-prijs voor het beste artikel in twee jaargangen van *Aquatic Ecology*.
- Simruralis: k€ 50 voor het ontwikkelen van een interactief computer spel om betrokkenen te leren over elkaars standpunten met betrekking tot landgebruik, vooral samen met Wim de Winter (toen *WU*, nu *WUR-WUR-Alterra*), maar ook Arnold Bregt (*WU*/*WUR-Alterra*), Ron van Lammeren (*WU*) en vele anderen.
- EVA: k€ 25 om de onderlinge competitie tussen mensen en scholeksters te modelleren bij het vangen van kokkels, samen met Wim de Winter (toen *WU*, nu *WUR-Alterra*).
- AMEPS: k€ 0 (het gedachtegoed werd door een deel van de wetenschappers goed ontvangen, maar door 'Brussel' niet gesubsidieerd) om een *Advanced Modeling Environment for Problem Solving* te ontwikkelen. Ondanks de geringe waardering heb ik toch veel geleerd van Marek Makowski (*IIASA*, Oostenrijk), Andrzej P. Wierzbicki (*Warsaw University of Technology*, Polen en *Japan Advanced Institute of Science and Technology*, Nomi, Japan), Janusz Granat (*Warsaw University of Technology* en *National Institute of Telecommunication*, Polen), Mietek Brdys (*University of Birmingham*, Engeland), Hans-Jürgen Sebastian (*RWTH Aachen*, en *GTS GmbH*, Duitsland) en Adrie Beulens (*WU*).
- Harmoni-CA: k€ 20 om een leidraad te schrijven over *Kwaliteitsborging van op modelleren gebaseerd waterbeheer* en om van MoST ProST te maken.

Al met al leverden deze projecten en projectjes niet alleen substantiële financiering op voor onderzoek van de leerstoelgroep, maar ik leerde er ook veel van. Een deel van het geleerde vond zijn neerslag in ruim 80 publicaties waarvan ik auteur ben of medeauteur. De breedheid van de onderwerpen maakt het bijna onmogelijk om 'een nietje te slaan door een aantal artikelen' en dat als proefschrift te beschouwen. Dat had alleen gekund door één van de vele onderwerpen uit de lijst hierboven te kiezen en de rest weg te laten. Maar dat was mijn eer te na. In dit boek heb ik geprobeerd een en ander met elkaar in verband te brengen.

Om van het moeras van projecten en onderwerpen uit de lijst hierboven iets constructiefs te maken, moest nog veel werk worden verricht. Een ontologische benadering ligt niet voor de hand bij een mathematisch bioloog die werkt in de informatica. Ontologieën (letterlijk de *zijnsleer*, vrij vertaald: 'het zijn en het wezen van het seinwezen') stellen je wel in staat om ordening aan te brengen in wat dan ook (inclusief het moeras aan onderwerpen uit de lijst hierboven). Veel van wat ik al doende geleerd heb, is door een ontologische molen gehaald en geordend. Voor veel mensen zal dit nogal abstract zijn, maar enkele van de ontologische resultaten zijn al voor praktische toepassingen gebruikt.

Dat het zover is gekomen en alles in dit boek is beschreven, is aan velen te danken.

Op de eerste plaats wil ik natuurlijk iedereen bedanken die genoemd is in de lange lijst van projecten hierboven. Zonder jullie had ik dit niet kunnen schrijven. Verder wil ik al mijn collega's van de leerstoelgroep *Toegepaste Informatiekunde* bedanken voor de aangename sfeer en de ruimte die ik kreeg mijn gang te gaan in het onderzoek. De belangrijkste bijdrage aan het onderzoek is van Ayalew Kassahun die in woord (als medeauteur van lezingen en artikelen) en daad (ontwerp en implementatie van MoST/ProST en de modelleerkennisbank) van onschatbare waarde was en is. Ik wil hem en collega Sjoukje Osinga ook bedanken dat ze mijn paranimfen wilden zijn. Jan Ockeloen heeft alle afzonderlijke (pdf)bestanden aan elkaar geknoopt, waarvoor dank. Sinds kort werken we samen met de leerstoelgroep *Operations Research en Logistiek* in de cluster *Logistics, Decision and Information Sciences*. Onze samenwerking schept in het onderzoek een breder perspectief op het toepassingsveld en de te gebruiken paradigmata. Verder verhoogt het de arbeidsvreugde onder het motto 'hoe meer zielen hoe meer vreugd'.

De promotiecommissie wil ik bedanken voor hun inspanningen in een laat stadium van mijn promotietraject. Prof. dr. ir. Alexander Udink ten Cate was vroeger collega op de vakgroep Informatica en ook medeauteur van enkele publicaties. Prof. dr. Aad Smaal ken ik nog uit mijn Middelburgse BALANS-tijd van twintig jaar geleden. Beste Aad, ik heb je beter leren kennen in de periode dat je zelf ging promoveren en veel van waar we het toen over hadden, is nu ontologisch gestructureerd hier beschreven. I learned to know dr. Marek Makowski as the organiser of the *Complex Systems Modeling workshops* at the IIASA in Laxenburg during many years, but especially in the preparation phase(s) of successive AMEPS proposals that were regretfully unsuccessful (so far?). Samenwerking met prof. dr. ir. Jan Top hoop ik in de toekomst te realiseren.

Mijn promotor prof. ir. Adrie Beulens wil ik bedanken voor alle ideeën en discussies. Ik weet dat ik niet altijd de makkelijkste was om te begeleiden. Eigenwijs als ik ben, wil ik liever mijn eigen weg gaan. Daarom ook bedankt voor de ruimte die ik kreeg om dat te doen. Mijn co-promotor prof. ir. Maurice Elzas heeft in het laatste stadium heel veel puntjes op de 'i' gezet. Heel erg bedankt. Jouw betrokkenheid gaf me soms het gevoel een beetje de erfgenaam te zijn van veel van jouw grootse gedachtegoed.

Carmen, je was en bent niet alleen praatpaal, maar ook bron van inspiratie, verstrekker van orde in de chaos, van correcties op het 'boekje' en van handige tips om abstracte zaken te ordenen. Zonder jou was ik niet aan mijn dissertatie begonnen en had ik er nooit een punt achter kunnen zetten. Je bent mijn alles, mijn persoonlijke *upper ontology*.

Heteren, november 2008-11-09 Huub Scholten

Over de auteur

Huub Scholten werd op 3 november 1951 geboren in Wisch. Hij behaalde het Gymnasium β diploma aan het Dominicus College in Nijmegen. Hij studeerde biologie (met nadruk op theoretische biologie) aan de Universiteit van Amsterdam. Na zijn docentschap aan verschillende middelbare scholen, was hij werkzaam bij de volgende organisaties. Eerst bij de Dienst Getijde Wateren (nu RIKZ) in Middelburg als modelleur bij project Balans. Vervolgens bij het Delta Instituut voor Hydrobiologisch Onderzoek (nu Centrum voor Estuariene and Mariene Ecologie van het Nederlands Instituut voor Ecologie) in Yerseke (ontwikkelaar bij het project Ecolmod en modelleur bij het project EOS). Nu is hij werkzaam bij Wageningen Universiteit, waar hij als universitair docent allerlei vakken over modelleren geeft en heeft gegeven. Ook is hij begeleider van studenten in hun MSc- en PhD-fase. Hij heeft deelgenomen aan een groot aantal onderzoeksprojecten (zie bijvoorbeeld *Post Scriptum* op pagina 309). Nu neemt hij deel aan het project AquaStress, gefinancierd binnen 6^e kaderprogramma van de Europese Commissie.

Zijn professionele belangstelling gaat uit naar modelleren en simuleren (methodologie, kwaliteit van het proces), veelal toegepast op milieuproblemen, waterbeheer en ecologie. Hij is medeontwikkelaar van software om goed, beter of best modelleren te ondersteunen. Verder gaat zijn belangstelling uit naar ontologisch structureren van kennis, zoals over modelleren en simuleren (zie dit proefschrift). Hiervoor is ook belangstelling vanuit andere toepassingsvelden getoond zoals slagveldsimulatie, postlogistiek en het ontwikkelen van nieuwe medicijnen in de biochemische en farmaceutische industrie. Zijn grote belangstelling voor hiermee samenhangende onderwerpen, zoals het kalibreren van wiskundige modellen, onzekerheidsanalyse en validatie.

Hij is als eerste auteur of co-auteur betrokken bij bijna 100 publicaties over het ontwikkelen en toepassen van domain specifieke modellen (ecosysteem, ecofysiologie, waterkwaliteit, enz.) en over methodologie en ondersteunende tools hiervoor. Voor enkele van zijn publicaties kreeg hij een prijs.

Een selectie uit deze publicaties:

Scholten, H. and M.W.M. Van der Tol, 1994. SMOES: a Simulation Model for the Oosterschelde EcoSystem. Part II: calibration and validation. Hydrobiologia 282/283, 453-474.

Scholten, H. and A.C. Smaal, 1998. Responses of Mytilus edulis L. to varying food concentrations - testing EMMY, an ecophysiological model. J. Exp. Mar. Biol. Ecol. 219, 217-239.

Scholten, H., Smaal, A.C., 1999. The ecophysiological response of mussels in mesocosms with reduced inorganic nutrient loads: simulations with the model EMMY. Aquatic Ecology 33, 83-100.

Refsgaard, J.C., H.J. Henriksen, B. Harrar, H. Scholten and A. Kassahun, 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software 20, 1201–1215.

Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.