Aligning Business Processes and IT of Multiple Collaborating Organisations

Ayalew Kassahun

Aligning Business Processes and IT of Multiple Collaborating Organisations

Ayalew Kassahun

Thesis committee

Promotors
Prof. A.J.M. Beulens
Emeritus Professor of Information Technology
Wageningen University & Research

Prof.Dr. B. Tekinerdogan
Professor of Information Technology
Wageningen University & Research

Other members
Prof.Dr M. Chaudron, Chalmers & Gothenburg University, Sweden
Prof.Dr P. Avgeriou, University of Groningen, The Netherlands
Prof.Dr M. Aksit, University of Twente, The Netherlands
Prof.Dr A.K. Bregt, Wageningen University & Research,

This research was conducted under the auspices of the Wageningen School of Social Sciences (WASS)

Aligning Business Processes and IT of Multiple Collaborating Organisations

Ayalew Kassahun

Thesis
submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus,
Prof. Dr A.P.J. Mol,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Tuesday 20th of June 2017
at 1:30 p.m. in the Aula.

A. Kassahun
Aligning Business Processes and IT of Multiple Collaborating Organisations
182 pages.

PhD thesis, Wageningen University, Wageningen, NL (2017)
With references, and summary in English

ISBN: 978-94-6343-210-8
DOI: http://dx.doi.org/10.18174/414988

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 BACKGROUND

Organisations today rarely operate in isolation but must collaborate in order to stay competitive. They collaborate in various ways, such as resource sharing, information sharing, outsourcing, collaborative design, collaborative planning, and collaborative operations. The various ways of collaboration can be grouped into strategic, tactical and operational collaboration. Successful collaboration requires alignments at all levels. Strategic collaboration requires the alignment of long term objectives and it may entail substantial redesign of entire business processes of the involved organisations (Hammer 1990, Davenport and Stoddard 1994). Tactical collaboration requires the alignment of mid-term goals. An example of tactical collaboration is collaborative planning (Barratt 2004, Kilger *et al.* 2015). Operational collaboration requires the alignment of cross-organisational business processes (Hoffner *et al.* 2000, Schulz and Orlowska 2004, Rezaei *et al.* 2014).

The alignment problems stated above are not new but have so far been addressed mainly from an individual organisation perspective (Henderson and Venkatraman 1993, Chan and Reich 2007). In this thesis we focus on collaborations across multiple organisations. We will specifically focus on operational collaborations which have become very relevant in the last decade due to their increasing complexity. We shall define the concept of *alignment* explicitly in this new context in chapter 3, but for the time being, it suffices to say that alignment is a state of agreement among business processes that span the collaboration network and the IT systems that support them; we refer to it as *business-IT alignment of multiple organisations* or simply *business-IT alignment*.

Realizing business-IT alignment is hard. It requires consideration of various potentially conflicting perceptions of the collaborating organisations on how the collaboration should be realized. When only few organisations are involved, realizing alignment is relatively manageable because the number of pair-wise alignments is limited. When multiple organisations are involved, alignment requires making complex agreements among the collaborating organisations. Generally it takes a considerable effort for two collaborating organisations to align their business processes and IT systems; doing so for many collaborating organisations is substantially harder.

Business-IT alignment is a necessary prerequisite for interoperability, and interoperability is a necessary prerequisite for collaboration. A strategy commonly adopted to achieve business-IT alignment—and to enable interoperability and collaboration—across a collaboration network of organisations is through compliance with a *reference architecture*[1]. A reference architecture is generally a specification, or a formal standard, that defines generic business process and IT models. A business process model defines the interrelationships of events, activities, decision moments, actors and information that collectively deliver an outcome for a client (Dumas *et al.* 2013). A reference architecture aims at enabling cross-organisational business processes by making the processes interoperable. IT systems are generally defined by their software architecture (ISO/IEC/IEEE 2011). Various types of models are used to define a software architecture but only a limited set of model types are used in reference architectures. An IT model of a reference architecture can be described as a model that specifies the IT services, defines the providers, brokers and users of the IT services, and

[1] The term *reference model*, instead of reference architecture, is also used in specifications and literature.

describes the relationships of the IT services with the cross-organisational business processes the IT services support. In order to be able to collaborate, organisations have to comply with the business process and IT models defined in the chosen reference architecture.

Generic reference architectures are commonly defined by global standardization bodies. The most influential standardization bodies in this respect include *Organisation for the Advancement of Structured Information Standards* (OASIS 2017), *Object Management Group* (OMG 2017), *Global Standards 1* (GS1 2017), *The Open Group* (The Open Group 2017), *American Production and Inventory Control Society* (which includes SCC-Supply Chain Council- APICS 2017), *International Organisation for Standardization* (ISO 2017) and *UN Centre for Trade Facilitation and E-Business* (UN/CEFACT 2017).

Examples of well-recognized specifications (or cluster of specifications) from these standardization bodies include the Service-Oriented Architecture (SOA, Erl 2008) related standards from OASIS, Electronic Data Exchange (EDI, Mukhopadhyay *et al.* 1995) from UN/CEFACT, the Supply Chain Operations Reference (SCOR, SCC 2012) from SCC and Electronic Product Code (EPC, GS1 2015) related standards from GS1.

Sector specific reference architectures are often derived from generic reference architectures. For instance, in the agri-food sector, which is one of the two focus areas of this thesis, several reference architectures have been developed based on SOA, SCOR and GS1 reference architectures (Steinberger *et al.* 2009, Verdouw *et al.* 2010, Wolfert *et al.* 2010, Kruize *et al.* 2016).

Ideally business-IT alignment is achieved using a generic reference architecture provided by a global standardization body. Organisations comply with a reference architecture by deriving their *concrete architectures* from a common reference architecture (Cloutier *et al.* 2010, Angelov *et al.* 2012). In practice, the organisations already have an information system in place and compliance is achieved by adapting the existing information system. Figure 1-1 shows this ideal scenario.

Unfortunately, a generic reference architecture is seldom good enough for at least two reasons. First, the reference architecture may not address the specific requirements of the sector it will be applied to. Second, the collaborating organisation may not be able to comply with it. In some cases there may be no suitable reference architecture and, as a consequence, a new one may have to be developed. This leads to three scenarios of applying a reference architecture as shown in Figure 1-2. Scenario ① is the ideal scenario presented in Figure 1-1. In scenario ② there is a suitable reference architecture that the collaborating organisations agree with but it is not sufficient and has to be adapted in order to address their specific concerns. In scenario ③ there is no suitable reference architecture and one will have to be developed.

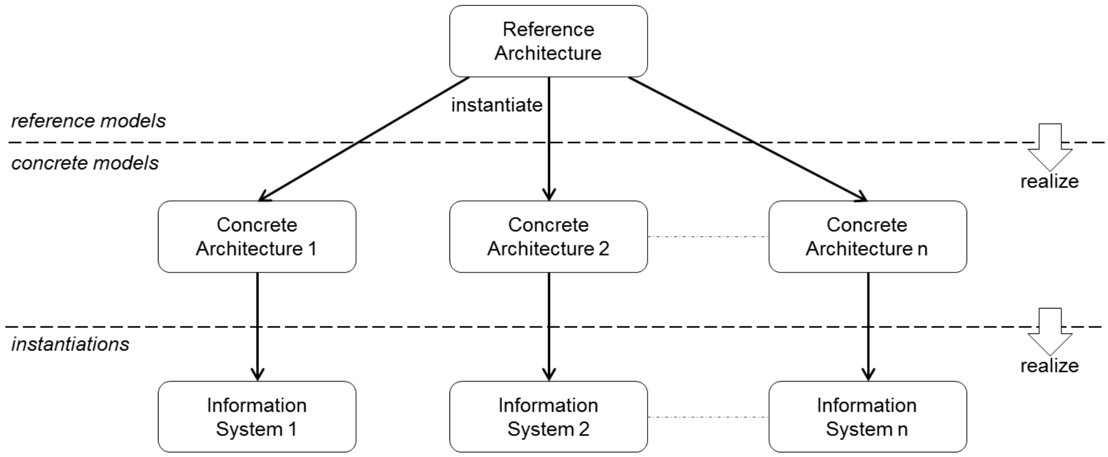


Figure 1-1: Deriving concrete architectures and the corresponding information systems from a reference architecture.

We present in this thesis an alignment framework for deriving a reference architecture. A new reference architecture has to address the specific concerns of the collaborating organisations. We identify two scenarios for designing a new reference architecture. These are shown as scenario ② and scenario ③ in Figure 1-2. In both cases the alignment framework helps derive a new reference architecture by aligning the business process and IT models adopted by the collaborating organisations. The question then arises: How to align models from diverse and multiple organisations?

Valuable insight can be gained from standardization processes standardization bodies use in deriving reference architectures. Standardization bodies, such as OASIS, ISO and GS1 follow long consultation processes with a large number of stakeholders who use case scenarios to describe their requirements. Ideally, the stakeholders should describe their requirements by presenting formal designs of concrete architectures they would like to realize, but formal designs are often not made available when stakeholders are engaged in deriving reference architectures. Also, ideally, standardization process should start from an existing generic reference architecture instead of from scratch. The standardization process that leads to a reference architecture is depicted in Figure 1-3. (Greyed dashed blocks are used to indicate that the desired generic reference architecture and concrete architectures may not always be available.)

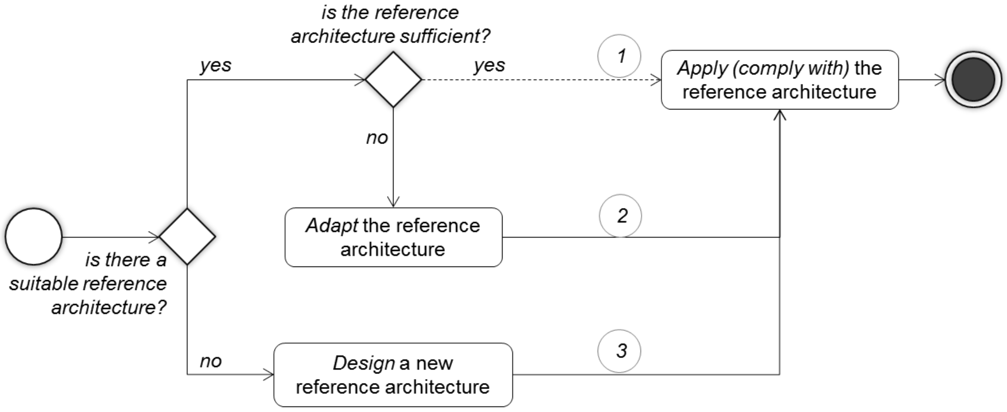


Figure 1-2: Three scenarios of applying a reference architecture.

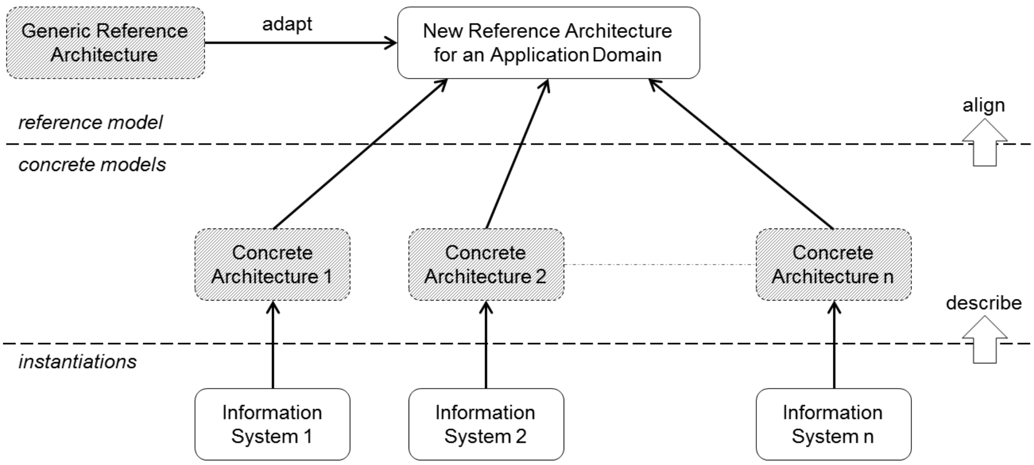


Figure 1-3: Deriving new reference architecture from existing reference and concrete architectures.

A reference architecture can provide various types of models but the most common are business process and IT models. A reference architecture for collaboration considers not the details of internal business processes, but business processes that span multiple organisations. Likewise, the IT models do not describe the details of IT systems deployed at the organisations, but the distribution of IT services across the collaboration network. In this thesis we also consider explicit models for representing the relationships between distributed IT and cross-organisational business processes. We refer to these three types of models in this thesis as *business collaboration models*. Thus, we refer to the business process models as *business collaboration process models*. We refer to the IT models as *business collaboration IT models*. We refer to the models that capture the relationships between the elements of business collaboration processes and distributed IT as *business collaboration process-IT models*. We particularly use the singular term *distributed IT system* (instead of distributed IT systems) in order to highlight the fact that the IT systems together provide new features that are not available when the IT systems are considered individually (Dahmann *et al.* 2008, Nielsen *et al.* 2015).

We identify three dimensions to the business-IT alignment problem of multiple collaborating organisations corresponding to the three types of business collaboration models. First, like the business-IT alignment problem inside an organisation, business processes have to be aligned with the underlying IT system. However, in this case, the business processes are cross-organisational. Likewise, the underlying IT system is distributed across the collaborating organisations. Second, the business processes of the collaborating organisations have to be aligned. Third, the IT systems of the collaborating organisations have to be aligned as well. We abbreviate the three types of alignment as BP2IT, BP2BP and IT2IT: BP2IT refers to Business Process to IT (or IT to Business Process) alignment; BP2BP refers to Business Process to Business Process alignment; and IT2IT refers IT to IT alignment.

Aligning requires comparing models to identify where misalignments occurred. In literature, models are often compared pairwise. This means that models adopted within one organisation are compared with the corresponding models adopted in another organisation. Complex analytical methods for comparing business process models pairwise are available (Weidlich *et al.* 2011, Becker and Laue 2012, Dijkman *et al.* 2013). We use a different strategy for comparing models. We adopt a reference architecture with which we compare multiple business collaboration models with one another.

To compare the organisations' view of the entire collaboration network with the help of business collaboration models, we need explicit models that can be compared with one another. However, practical and explicit modelling abstractions to do so are lacking. The available modelling abstractions that we can use to represent business collaboration models are difficult to compare with one another. Existing modelling abstraction for representing business process and IT models (including business collaboration models) include the Unified Modelling Language (UML, OMG 2015), Business Process Model and Notation (BPMN, ISO/IEC 2013) and the Architecture Modelling Language (ArchiMate, The Open Group 2013). These modelling abstractions are generally graphical and designed to enhance understanding but less suited for comparing two versions of the same model. The ability to compare business collaboration models is, however, essential for alignment.

In this thesis we provide models and a systematic approach for comparing business collaboration models as an alignment design framework. The framework is called BITA* (pronounce *bita-star*). *BITA* stands for Business-IT Alignment; the '*' denotes that multiple organisations are involved. The framework consists of three coherent design viewpoints (ISO/IEC/IEEE 2011) corresponding to the three types of business-IT alignment problems we described earlier. The viewpoints consist of modelling abstractions needed by the particular type of stakeholder concerned with the specific type of alignment.

Providing an alignment framework is, however, only part of the solution. The other major challenge has been elaborating how the alignment problem manifests itself in multi-organisational collaboration. The business-IT alignment literature has focussed to date mainly on the individual organisation and this aspect of the alignment problem has widely been researched from management perspective (Henderson and Venkatraman 1993, Chan and Reich 2007, Aversano *et al.* 2016, Hinkelmann *et al.* 2016). The alignment problems that occur when multiple organisations collaborate are, however, not well explored. To elaborate the alignment problem we consider two real-life case studies. The first shows how we can adapt an existing generic reference architecture to derive a new application area specific reference architecture. In the second we show how to derive an entirely new reference architecture.

1.2 Objectives

Alignment and reference architecture are the two main themes of this thesis. The first aim of this thesis is to further explore the nature of the alignment problem we aim to address. The other objective is to derive an alignment design framework and apply it in real-life business cases. Based on these objectives we derive the following research questions (*RQs*).

RQ1: How do alignment problems manifest themselves when multiple organisations collaborate?

Business-IT alignment problems have been explored in various ways but largely within the context of an individual enterprise. When multiple organisations are involved alignment problems have, to some degree, been addressed but from different perspectives, such as business process outsourcing, business process compliance, information system integration and interoperability, but have so far not been explored comprehensively and systematically.

To address the research question we explore the alignment of business processes and IT in real-life business collaboration cases.

RQ2: What are required modelling abstractions for aligning business-IT concerns of collaborating organisations?

To collaborate successfully, organisations adopt a common reference architecture to make their business process and IT systems interoperable with their collaboration partners. However, existing generic reference architectures often need adaptation to address specific requirements of a given problem context. Adapting an existing generic reference architecture or deriving a new one requires aligning business collaboration models adopted by the collaborating organisations. To address alignment problems we propose a business-IT alignment design framework. To define an alignment framework we identify relevant viewpoints and define modelling abstractions for each viewpoint. In addition we formulate a systematic approach for applying the modelling abstractions.

RQ3: How to apply the business-IT alignment framework and validate its utility in realizing collaboration systems?

The alignment framework must enable the alignment of business processes and IT across multiple organisations and thereby help design reference architectures. The ultimate goal is to help implement collaboration systems. To realize this goal we apply the framework to derive reference architectures and show how the reference architectures help realize collaboration systems. We demonstrate the utility of BITA* in two case studies.

1.3 APPROACH

This thesis uses design science research methodology following Hevner *et al.* (2004). Design science constitutes three research cycles, which are: (1) justifying the relevance of the design problem and identifying requirements for design through the *relevance cycle*, (2) designing the required design artefacts through the *design cycle*, and (3) grounding the design research in theory and in existing body of design knowledge through *rigor cycle.* Each of the three cycles requires an appropriate methodology.

The relevance cycle usually involves various requirements elicitation methodologies (Nuseibeh and Easterbrook 2000, Laplante 2013). In order to elicit the requirements the design problem and the environment where it arose should be relatively well understood, which is not always the case. When design problem and the environment are not well understood an empirical investigation of the problem in its real-life context is required. The business-IT alignment problem addressed in this thesis is such a case. To study the problem in its real-life context and justify its relevance we used a case study research methodology. Case study as a research methodology is widely used in social science studies such as psychology, sociology and political science but is also recently being applied in information systems engineering research (Easterbrook *et al.* 2008, Runeson and Höst 2008). We applied case study research to understand the problem context and to explore the nature of the alignment problems that arise in the given problem context.

The design cycle creates new design artefacts that have not existed before for addressing the problem at hand (Simon 1996, Hevner *et al.* 2004). To create the required design artefacts the design cycle uses an existing body of design knowledge (Hevner *et al.* 2004, Hevner and Chatterjee 2010). In this research we applied Business Process Modelling and Notation

(BPMN, ISO/IEC 2013), workflow patterns (van der Aalst and ter Hofstede 2011) and Service-Oriented Architecture (SOA, OASIS 2006) for designing business collaboration models. We also applied diverse other modelling abstraction, which are described where they are applied. The design contributions of this thesis are new design artefacts built with the help of the existing body of design knowledge. We will describe them using March and Smith's (1995) categories of design artefacts.

The rigor cycle ensures that the resulting design artefacts are valid; it also ensures that the design artefacts are not routine but new contributions to the body of design knowledge. The main contributions of this thesis in this respect are alignment modelling abstractions, reference architectures, and IT systems. To ensure the validity of the design artefacts we applied them in two case studies. The design artefacts are interdependent: the modelling abstractions are used to create reference architectures, the reference architectures are used to create information systems. We used the case studies to ensure that the modelling abstractions can, in fact, help design reference architectures, and the reference architectures can be realized in information systems. We discussed related literature to show that the design artefacts are new contributions to the body of design knowledge.

Case studies are used in order to justify the relevance of the problem and verify the resulting design artefacts. We used two problem domains that require multi-organisational collaboration and thereby also demonstrate the business-IT alignment concerns. In the first case study we used two large research projects in which a transparency system for meat supply chains has been designed and implemented. Realizing transparency in meat supply chains requires the collaboration of many organisations including farmers, slaughterhouses, meat processors and retailers. The meat sector has faced major crises and scandals despite the best efforts of many of the food operators to realize chain-wide transparency and avoid problems. We used the case study to both explore business-IT alignment problems and validate the resulting design artefacts. The second case study is a retrospective case study, also based on two large research projects. The term retrospective means here that the projects addressed business-IT alignment problems, though not explicitly, because they were conducted before this thesis project started. The projects were, in fact, conducted from 2002 to 2008 and were aimed at aligning environmental modelling processes and modelling IT systems. We used a retrospective case study because it is hard to organise a case study that involves a large group of organisations and substantial effort. The second case study is used to enhance the external validity of the study.

1.4 CONTRIBUTIONS

This thesis makes the following contributions:

- *Exploration of the alignment problem*

  We have demonstrated alignment problems that arise when multiple organisations collaborate. We have provided a detailed description of two case studies in which the alignment of collaboration processes and IT systems was a concern. We have described the stakeholders, their business processes and IT systems in details. We illustrated the alignment problems through examples consisting of conceptual, business process, and IT models.

- *Design framework for supporting business-IT alignment over multiple organisations*

We have developed a Business-IT Alignment framework called BITA*. The framework provides three explicit design viewpoints for three groups of stakeholders. The BP2BP alignment viewpoint is provided for process analysts for aligning the diverse business processes involved in business collaboration processes. The IT2IT alignment viewpoint is provided for software architects to align the underlying IT systems. The BP2IT alignment viewpoint is provided for an interdisciplinary team of the above two stakeholders to align the business processes with the underlying IT systems across the collaboration network.

- *Systematic approach for aligning business-IT alignment concerns over multiple collaborations*

We have provided a systematic approach for applying BITA* and demonstrated it in two real-life business cases. We showed how BITA* can be used to derive reference architectures for transparency systems and integrated environmental modelling systems.

- *Lessons learned and novel research directions*

The business-IT alignment problem of multiple collaborating organisations is novel. We have provided an alignment framework and reference architectures which can benefit both practitioners and researchers. We have shown how to apply workflow patterns to cope with business-IT alignment problems. Further, researchers can derive valuable insight and novel research directions, for instance, using design patterns in deriving reference architectures.

1.5 THESIS OUTLINE

The remainder of the thesis is organised as shown in Figure 1-4. The next chapter (Chapter 2) elaborates the research methodology that is briefly presented in this chapter. In chapter 3 we present the business-IT alignment framework BITA*. BITA* provides models for aligning business collaboration processes with the underlying distributed IT system and thereby assists

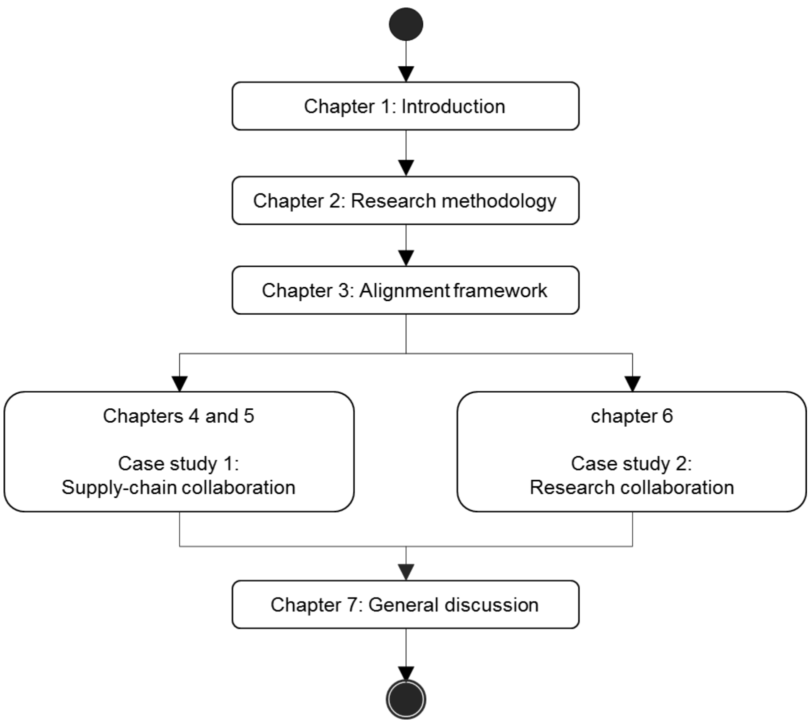


Figure 1-4: The outline of the thesis and reading guide.

the development of reference architectures. In chapter 4 we present a reference architecture for a transparency system of meat supply chains. This chapter also elaborates the business-IT alignment problems encountered when multiple and diverse organisation collaborate. In chapter 5 we present a transparency system that shows how a reference architecture can be applied in practice. In chapter 6 we apply the BITA* framework retrospectively to aligning modelling collaboration processes and IT systems and thereby facilitate the development environmental decision support systems. In chapter 7 we provide a general discussion and make concluding remarks.

2

The Research Methodology

2.1 INTRODUCTION

In this thesis we apply the design science methodology according to Hevner *et al.* (2004). Design science research follows three iterative cycles: *relevance cycle*, *design cycle* and *rigor cycle*. The relevance cycle motivates the desired improvement that should be brought about to an environment. It also leads to a list of requirements and associated criteria for evaluating the research results. The design cycle turns the requirements into new design artefacts using an existing body of design knowledge. The rigor cycle contributes to the body of design knowledge (Hevner 2007).

We applied case study methodology for the relevance cycle. We applied business process and relevant IT modelling techniques in the design cycle. We applied case study methodology, demonstration and review of related work for the rigor cycle. The remainder of this chapter is organised as follows. In section 2.2 we describe the design science methodology in the context of this thesis. In section 2.3, 2.4 and 2.5 we zoom in on the relevance, design and rigor cycles of the methodology, respectively. Finally, in section 2.6, we provide a summary of the methodology.

2.2 DESIGN SCIENCE METHODOLOGY

The general scheme of the methodologies adopted for this research is depicted in Figure 2-1. It shows how the motivation for the research and requirements identified in the relevance cycle, the design artefacts built in the design cycle, and the grounding of the results in the body of design knowledge in the rigor cycle are interrelated. The relevance cycle requires an environment that provides the context for the design cycle, derives requirements, and sets criteria with which the design artefacts are evaluated. The environment in which this research is conducted is a set of EU sponsored research projects which constitute the two case studies of this thesis. The design cycle is the heart of the design science research and it is where the design artefacts are created. The design cycle connects the relevance cycle, which provides the requirements, with the rigor cycle, which concerns grounding the research in theory and in existing body of design knowledge. The design artefacts can be categorized as constructs (conceptualizations), models, methods and instantiations (March and Smith 1995). We provided design artefacts in each of March and Smith's categories. We used the case studies, software implementation (proof by construction) and demonstrations to test if the resulting design artefacts can, in fact, address the problems they were designed to address. We presented background and related work to show that our contributions are novel.

2.3 THE RELEVANCE CYCLE

The relevance cycle of this thesis is formed by two sets of large European Commission sponsored real-life collaborative projects, which constitute the two case studies of this thesis. The projects provided us with the real-life cases. The case study research methodology is usually associated with social science research and is often contrasted with experimental research methodology in natural sciences. Designers traditionally identify requirements for designs through a process called requirements engineering (Nuseibeh and Easterbrook 2000, Wieringa *et al.* 2006). But when the problem is not well understood an empirical investigation of the problem in its real-life case studies may be required. In recent years case study research is accepted as valid research methodology in design science research (Easterbrook *et al.* 2008, Österle *et al.* 2011, Runeson *et al.* 2012). Case studies can be used in two ways: to explore and gather requirements or to validate the utility of design artefacts. The first case study is used for both purposes; the second case study is used mainly for the second purpose. We elaborate the case studies below.

2.3.1 Case Study 1: Requirements from Meat Sypply Chains

The first case study is used both to motivate the alignment framework and to identify the requirements for the framework. The case study was also used to demonstrate and validate the framework (see the rigor cycle in section 2.4).

The requirements for the framework are formulated in chapter 3 but they were also supplemented by literature and our experiences in developing collaboration systems in this and other cases. The development of the framework and the application of the framework went hand in hand. The requirements for the framework were identified during the development of a reference architecture for transparency systems in meat supply chains (chapter 4) and the realization of a transparency system based on the reference architecture (chapter 5). Transparency in meat supply chains involves many small and large food operators and third parties and provided an ideal case for demonstrating alignment issues encountered when multiple diverse organisations collaborate.

The case study was conducted in the context of two large research projects which were part of the Future Internet Public-Private Partnership (FI-PPP) research program of the European Commission (EC). The FI program was a six year program conducted from 2011 to 2016 and consisted of the FI-WARE core platform project and a number of use case projects that were executed in three phases. In the first phase eight use case scenario projects were sponsored to identify and analyse requirements of the different industry sectors on the FI-WARE core platform. In the second phase six use case trail projects were launched. The third phase

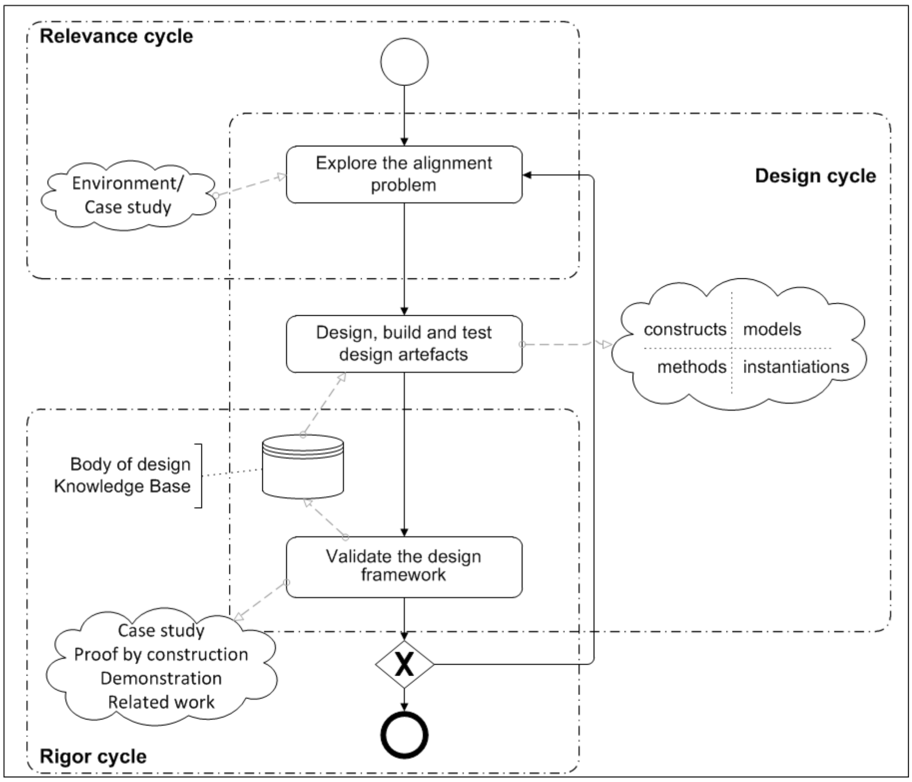


Figure 2-1: The research methodology of this thesis following Hevner et al (2004).

focused on innovation targeting start-ups and small and medium-sized enterprises (SMEs). The transparency system that was designed in connection with the case study has been realized by a start-up company (see chapter 4).

In the case study we were able to elaborate a design problem that is crucial in designing multi-organisational collaboration systems. We provided a conceptual model of a meat supply chain and elaborated the collaboration linkages of a series of food operators that transform slaughter animals into finished meat products. The food operators involved include, among others, farmers, a series of meat processors, and retailers. In meat supply chains many farmers and a number of meat processors can be involved. In the first case study we achieved two objectives. First, we elaborated the alignment problems and showed how and why they occur. Second, we validated the utility of the alignment framework we developed by actually realizing a reference architecture, realizing a transparency system using the reference architecture and demonstrating the resulting transparency system (see chapter 5).

2.3.2 Case Study 2: Requirements from Environmeantal Modelling

The second case study is used to enhance the validity of the research by showing the utility of the framework in a different type of multi-organisational collaboration setting.

The case study involves two past projects conducted from 2002 to 2008. The aim of the projects was to develop a methodology and toolbox to support collaborative environmental modelling studies. The first project was called HarmoniQuA. HarmoniQuA was part of the CatchMod (Catchment Modelling) program of the European Commission, which consisted of ten research projects initiated for supporting Europe’s Water Framework Directive (WFD, EC 2000). A number of CatchMod projects aimed at harmonizing the processes, models and methods of river basin modelling, an aim which is reflected in the names of the projects, such as HarmoniQuA and HarmoniRiB. In the HarmoniQuA project we developed a quality assurance procedure that consists of a multidisciplinary guideline and modelling support software systems that support the collaboration of researchers from different organisations and disciplines (Kassahun *et al.* 2007, Scholten *et al.* 2007). The guideline and the software system were tested in ten pilot modelling test cases in two rounds of testing. The second project was the AquaStress project, in which the software systems developed during the CatchMod programs were enhanced and integrated (Kassahun *et al.* 2008, Assimacopoulos *et al.* 2009).

In chapter 6 we describe the motivation for applying the alignment framework in this case study.

2.4 The Design Cycle

Design as a research approach is driven by the need to improve the current state of affairs by introducing new and innovative design artefacts (Gregory 1966, Simon 1996, Hevner *et al.* 2004). New design artefacts are developed using other design artefacts from the existing body of design knowledge base—in the same way that new tools are built using existing tools. In section 2.4.1 we describe the relevant existing design knowledge we applied. In section 2.4.2 we describe the new design artefacts that are presented as alignment design framework and which constitute our contribution to the body of design knowledge base.

2.4.1 Existing Design Artefacts

Earlier we have mentioned that different organisations adopt different business collaboration models. We describe in this section which design artefacts are available in the existing body of design knowledge base for designing business collaboration models.

A widely used method for modelling business collaboration processes is Business Process Modelling and Notation (BPMN, ISO/IEC 2013). A business process is essentially a set of activities that are triggered by other activities, gateways or events (Dumas *et al.* 2013, ISO/IEC 2013). In BPMN various types of events, tasks and gateways are defined. The most prominent elements of BPMN models are start and end events, tasks (activities), and exclusive choices, inclusive choices, parallel flow gateways, lanes and pools. A start event signals the start of a process, and an end event the end of the process. A task represents an atomic piece of work. A gateway defines how the sequence flow branches and merges. A parallel fork and join gateway represents concurrent sequence flows; inclusive and exclusive choice gateways represent a decision of choosing one set of activities over another. Collaboration within integrated business units is modelled by grouping activities in *lanes*. Collaboration across organisational boundaries or across loosely integrated business units of an organisation is modelled by grouping activities and lanes in *pools*. Each lane and pool identifies the organisation or business unit that performs the activities. In this thesis we use pools to represent organisations. The interactions among organisations are represented as message exchanges. These are just part of many other modelling abstractions of BPMN (Ouvans *et al.* 2006, Chinosi and Trombetta 2012).

Apart from BPMN, other business process modelling notations in use include IDEF[2] (Cheng-Leong *et al.* 1999), Event-driven Process Chain (EPC, Scheer 1992) and activity diagrams of the Unified Modelling Language (UML, OMG 2015). These methods are mainly graphical notations optimised for human understanding and communication, but less suited for analysis. Modelling abstractions that are well suited for analysis include Petri-nets (van der Aalst 1998), pi-calculus (Sangiorgi and Walker 2003) and Guard-Stage-Milestone (GSM, Hull *et al.* 2011). Though the various modelling abstractions are not equivalent, they can largely be translated from one to another form (Ouvans *et al.* 2006, Dijkman *et al.* 2007, Wong and Gibbons 2008).

Another approach to representing business collaboration processes is using workflow patterns. Workflow patterns are names assigned to recurring snippets of business processes or aspects of the related IT. The most prominent categories of workflow patterns are *control-flow*, *data-flow* and *resource-flow* patterns. Control-flow patterns can be used to describe complex business processes using only few control flow patterns, which are basically the names of the patterns. Dataflow patterns can be used to data sharing between the IT systems that support business processes. Likewise, resource flow patterns describe the patterns of resource allocations in business processes (van der Aalst and ter Hofstede 2011). Workflow patterns are used far less in practice. In this thesis we apply control-flow and dataflow patterns to facilitate alignment.

[2] IDEF=Icam DEFinition; Icam=Integrated Computer Aided Manufacturing

When it comes to modelling IT, several types of models are used. The basic design artefact that shapes the various designs is presented as a software architecture. However, a software architecture consist of various designs from different perspectives, called views (Clements *et al.* 2010). The various views collectively identify the components of the IT system, the interaction among the components, and the interaction of the system as a whole with its environment (ISO/IEC/IEEE 2011).

An IT system is generally designed within the context of an organisation that deploys it. A collaboration system, however, consists of IT systems that are distributed across the collaborating organisations. Though the IT systems are owned by different organisations, together they should provide the functionalities that are required to realize collaboration, and as such, they are not a simple collection of IT systems but a *system of systems* (Ackoff 1971, Boardman and Sauser 2006). We, therefore, refer to the IT systems collectively as *distributed IT system* (using a singular form) to highlight the required integration.

Generally there is no central authority that designs a distributed IT system and implements it. Instead, organisations agree with a reference architecture that they can comply with and that enables the integration of their IT systems. There are many generic reference architectures, such as SOA (OASIS 2006) and EDI (Mukhopadhyay *et al.* 1995), which constitute part of the existing body of design knowledge base, and which can be used to create the desired integration.

The broadly-accepted approach of representing distributed IT systems is the SOA. SOA enables modelling the distribution of IT systems by providing the required modelling abstractions for representing the distribution of IT systems and information exchange among them. The distribution of IT systems is modelled using one of the three SOA roles: service consumer, service provider or service broker (OASIS 2006). Various specifications are defined to facilitate and define information exchange, including SOAP (Mitra 2003), WSDL (W3C 2007), UDDI (OASIS 2004) and REST (Fielding 2000).

2.4.2 New Design Artefacts

So far we have presented in brief existing modelling abstractions that are available to us for modelling business processes and IT systems. The main purpose of this thesis is to add to the existing body of design knowledge, and thus to add modelling abstractions to support business-IT alignment and provide a systematic approach for applying them. Below we describe how new design artefacts are organised and described.

In software engineering literature a consistent set of modelling abstractions is referred to as a design framework. Unfortunately, the term framework is defined and redefined by framework providers and “the world has not really settled on a precise definition” (Schekkerman 2004). The concept of a framework was first used in (or at least popularized by) the Zachman architectural framework. The Zachman framework has been developed for the purpose of depicting the structural and behavioural models of a complex system in a simple matrix structure (Zachman 1987). While the Zachman framework is extended and is still in use (Zachman 2016), The Open Group Architecture Framework (TOGAF, The Open Group 2011) have become the de-facto standard enterprise architecture framework. It is important to notice here that both Zachman and TOGAF, and most other frameworks, primarily focus on the enterprise[3], i.e. the individual company or business organisation.

The framework we will present is focused on multiple organisations. Many frameworks are designed to be applied within an enterprise, and are also referred to as enterprise frameworks. Many others are referred to as architectural frameworks and focus of software architecture. To emphasize that our framework is not limited to enterprise and also not to software systems, we refer to our framework as a *design framework*—specifically an *alignment design framework.*

A framework consists of different viewpoints, each viewpoint focusing on specific group of design stakeholders (designers). This approach to design is similar to designing a building architecture in the construction industry. A building is described by site plans, floor plans, elevation views, and various cross-section views. In software engineering a viewpoint is designed using the ISO/IEC/IEEE (2011) standard. The standard provides an extensible metamodel for defining viewpoints, which we adopt to describe the viewpoints of our design framework.

In software engineering a number of viewpoints have been formally identified and designed (Kruchten 1995, Hofmeister *et al.* 2000, Kruchten 2004, Lattanze 2008, Clements *et al.* 2010). Modelling abstractions outside the software engineering field are not customarily referred to as viewpoints. In this thesis, though, we refer business process modelling using BPMN and distributed IT modelling using SOA-related modelling abstractions as viewpoints. Both BPMN and SOA related specifications qualify as viewpoints following the ISO/IEC/IEEE metamodel. In both cases there are a clear set of stakeholders with specific design concerns that the modelling abstractions address.

Table 2-1: The elements of a viewpoint.

Viewpoint Element	***Description***
Name	The name of the viewpoint
Stakeholders	The types of stakeholders involved.
Concerns	The concerns the stakeholders address with the help of the viewpoint.
Related viewpoints	The related viewpoints, particularly, the viewpoints which share modelling constructs
Model kinds	
Elements	The model elements or constructs
Relations	The relations among the model elements
Attributes	The attributes of elements and relations
Notation	The notation used to represent the models

Table 2-1 provides the elements of a viewpoint according to the ISO/IEC/IEEE metamodel. A viewpoint describes *models kinds* (modelling abstractions) used in the viewpoint. To define new modelling abstractions a viewpoint generally uses other modelling abstractions from existing viewpoints. To use Hevner's terminology new design artefacts use design artefacts from the existing body of knowledge. New modelling abstractions are created from existing

[3] Enterprise means "a business or company" according to the Oxford dictionary, and "a business organisation" according to the Merriam-Webster dictionary.

modelling abstractions by defining new relationships among the existing abstractions or incorporating new attributes.

The viewpoints defined in this thesis are all alignment viewpoints. We defined viewpoints and associated modelling abstraction for the three types of alignment concerns mentioned in the introduction of this thesis, namely BP2BP, BP2IT, and IT2IT viewpoints. We provide the elements, relations, attributes and notations for the modelling abstractions of the viewpoints.

2.5 THE RIGOR CYCLE

The rigor cycle ensures the design artefacts represent innovations and thus contributions to the body of knowledge (Hevner 2007). The design artefacts can be new constructs, models, methods, and instantiations (March and Smith 1995). In the rigor cycle we use case studies, implementations and demonstratins. To ensure the validity of the design artefacts we instantiate and demonstrate them in practice, providing "proof by construction" (Nunamaker *et al.* 1990, Hevner *et al.* 2004). Instantiations are further evaluated either through qualitative or quantitative measures, such as functionality and user friendliness (Peffers *et al.* 2008). We adopt an iterative approach in which no single evaluation step is done but evaluations are part of the requirements gathering process (Sein *et al.* 2011). The design, implementation and demonstration iterations are performed until a satisfactory solution is achieved within the limits of the scope, budget and time of the projects in which the designs are made.

2.6 SUMMARY

In this thesis we applied the design science research methodology following Hevner *et al.* (2004). The methodology consititutes three research cycles: relevance cycle, design cycle, rigor cycle. Each of the three cycles requires an appropriate methodology. We used case study methodology in the relevance cycle. The relevance cycle requires an environment that provides the context for the research. Four large EU sponsored research projects provide the context for the research. Two of the projects are about transparency in meat supply chains and constitute the first case study. The other two projects are about collaboration in environmental modelling studies constitute the second case study. The second case study is performed retrospectively since the projects were already completed before the thesis research started. In the design cycle we applied business process modelling, workflow patterns and the service-oriented architecture approach. The design cycle resulted in design constructs, models, methods and instantiations. The constructs are used to conceptualize design models. The models are organised in viewpoints. The viewpoints constitute the alignment framework. The framework is used to create new collaboration design artefacts. In the rigor cycle we used case studies, "proof by construction" and demonstrations to ensure the validity of the new contribution to the existing body of design knowledge base.

3

BITA*: Business-IT Alignment Framework of Multiple Collaborating Organisations[1]

[1] Submitted to the journal of Business & Information Systems Engineering (ISSN: 1867-0202): Kassahun, A. and Tekinerdogan, B. (submitted). BITA*: Aligning Business Processes and IT of Multiple Collaborating Organisations based on Reference Architecture.

Abstract

Businesses today rarely operate in isolation but must collaborate with others in a coordinated fashion. For supporting collaboration, it is important that business processes and the supporting IT systems of participants are well aligned. Although the business-IT alignment problems have been broadly addressed in the literature the focus has been mainly on alignment within an organisation, and less attention has been given to the alignment across multiple organisations. Specifically, the alignment of business collaboration processes and the supporting distributed IT system has not been addressed systematically. In this paper, we propose BITA*, which is a business-IT alignment framework for multiple and diverse collaborating organisations. BITA* includes both the required alignment modelling abstractions as a coherent set of design viewpoints, as well as a systematic approach for applying the modelling abstractions. BITA* provides three viewpoints to be used by three groups of stakeholders. The BP2BP alignment viewpoint is used by business analysts to align diverse business collaboration process models. The IT2IT alignment viewpoint is used by software architects to align diverse models for the distribution of data and IT systems. The BP2IT alignment viewpoint is used by an interdisciplinary team to align diverse models for supporting the business collaboration processes with the distributed IT system. The application of the framework is demonstrated using a real-life business case.

Keywords: Business-IT alignment, Business collaboration processes, BPM, Reference architecture, Workflow patterns

3.1 INTRODUCTION

To achieve their business goals, businesses today rarely operate in isolation but must collaborate in a variety of processes with others in a coordinated fashion. For supporting the collaboration, it is important that the business processes and the supporting Information Technology (IT) are well aligned. This implies that the business process models of the different collaborating organisations are interoperable with each other to realize business integration.

In fact, business-IT alignment problems are not new and have been broadly addressed in literature. However, the problems have been mainly addressed within the context of a single organisation. Hereby, less attention has been dedicated to aligning the businesses processes that define the collaboration across different multiple organisations. It is true that orchestration and choreography languages have been proposed in the context of IT integration to support the executability of business processes across organisational boundaries (Peltz 2003, Newcomer and Lomow 2004, Erl 2008, Liu *et al.* 2009, Cummins 2015). But, the integration requires business processes and IT systems that are aligned in the first place. Unfortunately, the explicit design abstractions and the corresponding design heuristics for aligning misaligned business processes and the underlying IT systems of multiple collaborating organisations are largely missing.

Cross-organisational business processes and the underlying IT systems that are distributed across the collaboration network can be misaligned in three ways. First, the cross-organisational business processes that each individual organisation adopts for its collaboration with its partners (hereafter referred to as *business collaboration processes*) may be misaligned. We call this concern Business Process to Business Process *(BP2BP) alignment concern*. Second, each organisation has its own models on how the business collaboration processes should be supported by an integrated IT system composed of its own and its business partners' IT systems (hereafter referred to as the *distributed IT system*)—or, how the distributed IT system should be exploited by the business collaboration processes. The models of distributed IT support to business collaboration processes, and vice versa, that different organisations adopt may not agree with each other. We call this concern Business Process to IT *(BP2IT) alignment concern*. Third, the models governing how data and IT systems should be distributed and integrated to form the distributed IT system also differ from organisation to organisation. We refer to this concern as IT system to IT system *(IT2IT) alignment concern*.

Literature from various disciplinary backgrounds addresses the three alignment concerns often without explicitly formulating them as alignment problems and also not in a coherent manner. BP2BP alignment concerns have largely been addressed in management and business literature, such as business process outsourcing (Davenport 2005), business process compliance (Sadiq *et al.* 2007), and business process maturity (Rosemann and vom Brocke 2015). BP2IT alignment concerns have, in fact, gained much attention in the business-IT alignment literature but mainly within the limited scope of the individual business organisation (or the enterprise). From an IT perspective, BP2IT alignment concerns have often been seen as a one-way design problem where the IT design is considered to follow the business design (Wieringa *et al.* 2003, The Open Group 2011, Zachman 2016). BP2IT alignment concerns as a two-way alignment problem can be addressed by modelling the relationships between cross-organisational business processes and the distribution of IT systems across collaborating organisations. But, this aspect of BP2IT alignment has so far not been addressed systematically. IT2IT alignment concerns have largely been addressed as coupling, integration or interoperability issues (Chen *et al.* 2008, Daclin *et al.* 2016).

In this paper, we propose BITA*, which is an approach for aligning business collaboration processes and the underlying IT systems of multiple collaborating organisations. BITA* includes both the required alignment modelling abstractions as a coherent set of design viewpoints, as well as the systematic approach that is necessary for applying the modelling abstractions. BITA* provides three viewpoints to be used by three groups of stakeholders. The *BP2BP alignment viewpoint* is used by business analysts to align diverse business collaboration processes. The *IT2IT alignment viewpoint* is used by software architects to align diverse IT systems. The *BP2IT alignment viewpoint* is used by an interdisciplinary team of business analysts and software architects to align business collaboration processes with the supporting IT system that is distributed across the collaboration network.

To define and model alignment we introduce the concept of *allocation*. *Allocation* is a relationship matrix. Allocation is a representation of business collaboration models (which are generally given graphically) in a matrix form. Three types of business collaboration models are defined: Business Collaboration Process (BCM) model, Distributed IT (DIT) model, and model for describing the relationship of BCM and DIT (BCM-DIT) model. We also introduce the concept of *alignment* (so far used informally). *Alignment* is a direct comparison of two corresponding allocation matrices. Alignment refers to the idea that each organisation has its own business collaboration models, which are not necessarily aligned with those of its partners, and which need aligning. For every possible pair of organisations it is in principle possible to compare each allocation matrix of the first organisation with the corresponding allocation matrix of the second organisation. However such an approach to alignment involves *nC2* (n combination 2) possible comparisons, where *n* is the number of organisations. This is probably only acceptable for a small number of organisations. An alternative approach is to use a reference model. When a reference model is used the allocation matrices of each organisation are not compared with each other but with the corresponding allocation matrix of the reference model. Such an approach involves a maximum of *n* comparisons.

Still, the above approach to alignment is impractical because many organisations could not produce the required explicit business collaboration models that have to be compared with the reference model. The practical approach used in BITA* is to use the reference allocation matrices to describe the business collaboration models the organisations adopt. According to this approach, alignments are modelled by associating alignment attributes to an alignment matrix. The alignment matrix is derived from the reference allocation matrix and from informal descriptions of the different business collaboration models of the organisations. The informal descriptions, which are usually use-case stories and use-case scenarios, determine the alignment attributes that have to be filled in the alignment matrix. Alignment modelling is the core of the BITA* framework and is demonstrated in detail using a real-life business case.

The rest of the paper is structured as follows. In section 3.2 we provide the building blocks of the business-IT alignment framework. In section 3.3 we present the case study and formulate the problem statement. In section 0 we present the BITA* alignment design framework. In section 3.5 we apply the BITA* approach using the case study as a running example. In section 3.6 we discuss the application of the approach. In section 3.7 we present related work, and finally in section 3.8 we provide the conclusion.

3.2 Building Blocks

In the following we present the required building blocks of the business-IT alignment framework including business process models (section 3.2.1), workflow patterns (section 3.2.2), and IT models (section 3.2.3).

3.2.1 Business Process Models

Business process models (BPMs) are formal mechanisms for defining business processes. Originally introduced for modelling collaboration among functional departments of an organisation (Davenport and Short 1990, Hammer 1990, Harrington 1991), business process models are, nowadays, extensively used to model collaboration across organisational boundaries.

Probably, the most widely adopted approach for modelling business processes is Business Process Model and Notation (BPMN, ISO/IEC 2013). BPMN provides three types of models: process model (Chapter 10, ISO/IEC 2013), collaboration model (Chapter 9, ISO/IEC 2013), and choreography model (Chapter 11, ISO/IEC 2013). A process model (PM) describes the sequencing of activities within an organisation. Collaboration and choreography models are used to model business collaborations processes. A model of collaboration across organisational boundaries, in its simplest form, consists of pools across which messages are exchanged. A pool represents an organisation and may or may not include the PM of the pool. A choreography model describes the interactions (instead of message exchanges) among the collaborating organisations. A choreography model is a different form of representing a collaboration model.

3.2.2 Workflow Patterns

Business process modelling primarily focuses on how to represent the different process workflows. However, business process models generally contain many recurring elements that business process modellers often come across. These recurring problem-solution pairs are called workflow patterns (Russell *et al.* 2006). In the past, more than a hundred workflow patterns have been identified, categorized and catalogued (van der Aalst and ter Hofstede 2011). The most prominent categories are *control-flow*, *data-flow* and *resource-flow* patterns (van der Aalst *et al.* 2003). A Control-Flow Pattern (CFP) defines a recurring pattern of sequencing of activities in business process models. A Data Flow Pattern (DFP) models the patterns of data access and usage that are often encountered by business process and IT system modellers. Resource Flow Patterns (RFPs) define the patterns of resource allocations in business processes. In this paper we apply CFPs and DFPs only.

3.2.3 IT models

The basic artefact that shapes the design of IT systems is software architecture. Software architecture defines the components of the software system of an organisation, the interactions among the components, and the interaction of the system as a whole with its environment (ISO/IEC/IEEE 2011). It is particularly useful to guide the design and analysis of the system and support the communication among its stakeholders (Tekinerdogan, 2014).

Collaboration involves IT systems that are distributed across collaborating organisations (including nowadays IT service providers). The integration of the IT systems requires that they comply with a common specification, generally referred to as a reference architecture. A reference architecture guides the design of the concrete architectures of the collaborating organisations (Cloutier *et al.* 2010, Angelov *et al.* 2012). Hereby, concrete architecture refers to a software architecture for a specific context (*i.e.* for a particular organisation or set of organisations) and that which can be implemented into a software system.

A reference architecture for a distributed system is nowadays defined using the Service-Oriented Architecture (SOA) approach. According to this approach an IT system of a participating organisation is represented by the interfaces it provides. The simplicity (or complexity) of the distributed systems is therefore determined by the simplicity or complexity of the IT services and their interactions, and not by the complexity of the individual IT systems. In SOA, the collaborating organisations can take one or more of the following roles: *service client*, *service provider* and *service broker* (OASIS 2006). The desired integration is achieved by *publishing* the IT services in a *discovery* services to help clients find the IT services and their providers, and exchanging *messages* (data) based on standardized protocols (Barry 2003, Papazoglou *et al.* 2008, Buyya *et al.* 2009).

3.3 A Case Study and Problem Description

In the following we will describe a case study that we will use as a running example to illustrate the problem and the application of the framework. The case study has been applied in two large research projects conducted from 2011 to 2015 as part of the Future Internet (FI) program of the European Commission (EC). The case study concerns the adaptation of a generic transparency reference architecture to meet requirements of meat supply chains.

3.3.1 Case Study: Transparency System for Meat Supply Chains

A supply chain is a set of three or more entities that move products, services, finances, information, or any combination of these upstream to sources or downstream to customers (Mentzer *et al.* 2001). A meat supply chain consists of a network of food operators that transform slaughter animals into finished meat products. The input to the supply chain is provided by suppliers, who include breeders and feed suppliers, while the output is provided to customers, who are largely consumers. An important concern in meat supply chains is how to provide chain-wide transparency in order to meet the requirements of safety, quality and consumer trust in meat products (Kassahun *et al.* 2014). To meet these requirements a transparency system for meat supply chains needs to support the collaboration of the responsible supply chain actors, who have to supply transparency information. The responsible actors in meat supply chains can be categorized as food operators and third parties. Food operators include the farmers, meat processors, distributors and retailers. Third parties include regulators, inspectors, and laboratories. The collaboration involves the sharing of transparency data among the supply chain actors. A conceptual model for meat supply chain transparency systems is shown in Figure 3-1.

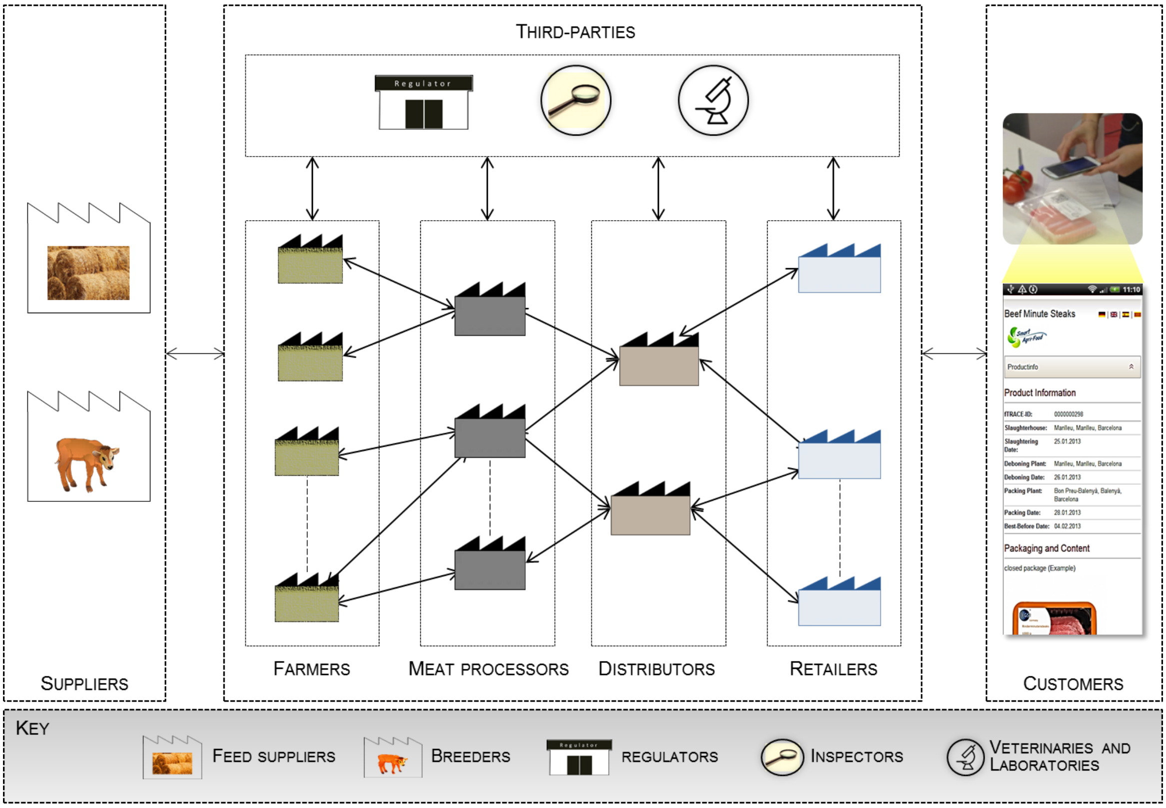


Figure 3-1: A conceptual model of meat supply chains

To achieve chain-wide transparency the relevant business collaboration processes and the distributed IT of the collaborating actors (food operators and third parties) have to be aligned. To achieve alignment at least two conditions have to be met: (1) there is a reference architecture that defines common business collaboration processes and IT models in sufficient details, and (2) all actors comply with the reference architecture. A recognized global standard that aims to achieve the first goal is the Electronic Product Code Information System (EPCIS, EPCglobal 2014). EPCIS is a specification based on SOA developed by GS1. GS1 is a global consortium that designs global standards for supply chain transparency including the numbering system for barcodes that are used in virtually every consumer product and logistic package. Achieving the second goal based on the EPCIS specification—with no modification—turned out to be infeasible for many supply chain actors. Before we describe the problems of compliance with the EPCIS standard that the second goal represents, we first describe the EPCIS specification.

EPCIS represents a generic reference architecture that applies to all supply chains. EPCIS specifies a distributed network of enterprise transparency systems that are loosely connected through a discovery service. EPCIS standardizes the *data capture* and *data query* processes, which are two key processes in any transparency system. The data capture process defines how transparency data should be scanned (or read by any other means) from each product item and stored in a transparency data repository. Here, the data primarily correspond to the events that are related to the physical movement or processing of products (such as loading, cutting and mixing). A *data query* process defines the queries for retrieving transparency data from transparency data repositories. According to the EPCIS specification data capturing is a local process that is carried out independently by each food operator, and *data querying* is a business collaboration process that involves multiple food operators and third parties. In order to distinguish between local processes, on one hand, and the integration of those local processes into a business collaboration process, on the other hand, distinction is made in the literature between *internal transparency systems* (*ITS*) that provide the ability to capture and query transparency data within an organisation, and *external transparency systems* (*ETS*) that provide the ability to query transparency data across the supply chain (Moe 1998, Gandino *et al.* 2009).

Different query business collaboration processes can be defined based on the EPCIS reference architecture (Kürschner *et al.* 2008, Lorenz *et al.* 2011, Kywe *et al.* 2012). Figure 3-2 shows an example of a query business collaboration process model for retrieving transparency data across meat supply chains. The model consists of four business processes that are executed locally by the respective organisation. PM_{ITS} and PM_{ETS} represent the data query process models (PMs) that take place at the food operators. The PM that receives the request for transparency data for the first time is a software application (*app*). The *app* implements the business process PM_{app}. The reference architecture does not specify who should provide such an application and based on our experience we assume it will be an external third party. The app triggers PM_{ETS} of a food operator. PM_{ETS} is provided by what we here refer to as the *focal* food operator—*focal* because it receives the request for transparency data on behalf of the supply chain. Also note that the term *focal* is not a permanent role but is only valid for the given request. According to the example query business collaboration process model the focal food operator realizes the external transparency by retrieving transparency data across the supply chain. It does so by querying transparency data locally and externally (from the transparency systems of other food operators) recursively. The subscript ETS indicates, therefore, that PM_{ETS} realizes external (chain-wide) transparency. PM_{ITS} retrieves transparency data only locally, from the local EPCIS repository (the subscript ITS indicates that PM_{ITS} realizes internal transparency).

The focal food operator 'discovers' the addresses of its partner food operators from a registry maintained by a third-party, which is not necessarily the same third party that provides the app. The discovery process is represented by the process model $PM_{discovery}$. Given an ID of a product item, the discovery service provides a list of *URLs* representing ITSs from where transparency data can be queried. In general PM_{ETS} and $PM_{discovery}$ can be complex. In practice, these business processes are complex processes for which diverse approaches are proposed (Kürschner *et al.* 2008, Lorenz *et al.* 2011, Kywe *et al.* 2012).

Finally, end-users use an app to scan the ID of a product item and retrieve associated transparency data about the product item. For simplicity we assume that each meat product item has a unique ID printed as barcode which end-users can scan. PM_{app} represents the business process implemented by such an app. This process combines the outputs from PM_{ETS} with product descriptions (retrieved from a master data repository) into understandable and user-friendly information and presents it to the user (for a detailed demonstration refer to Kassahun *et al.* 2016).

It is important to note that in the query business collaboration process the details of the internal business processes are not provided. Only the activities that are candidate for alignment are included because those activities can potentially be assigned to a different organisation (*i.e.* to a different pool). Such activities are the concern of the alignment effort.

3.3.2 Problem Description

For the given case description we have derived a business collaboration process from the reference architecture detailing the process models of the supply chain actors. The desired chain-wide transparency is realized when each supply chain actor supports the processes that

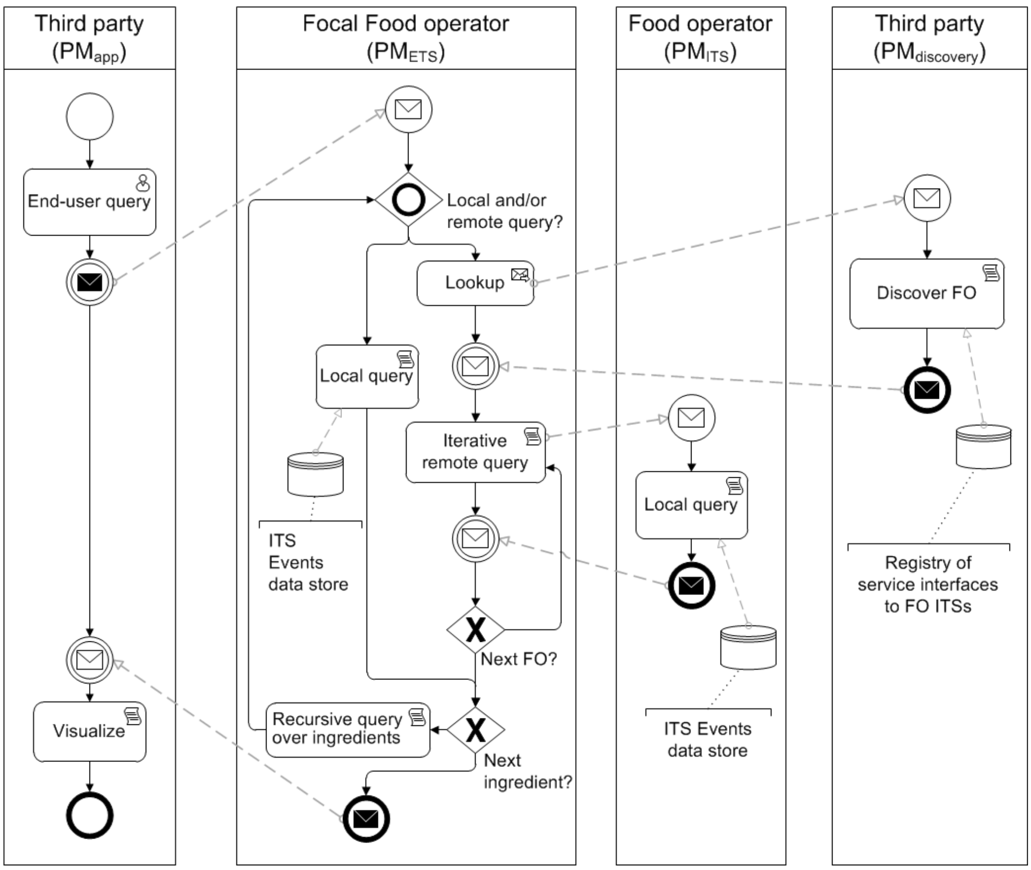


Figure 3-2: A query business collaboration process according to the EPCIS reference architecture

are assigned to it and thereby complies with the business collaboration process model.

Unfortunately, many of the supply chain actors do not, and cannot, comply with the generic EPCIS reference architecture and the business collaboration process models that can be derived from the architecture. There are three basic reasons for this. First, several European food regulations impose conflicting requirements on transparency systems. For instance, regulations on the movement and slaughter of bovine animals (EC 2000, EC 2004) mandate a different type of business collaboration than the General Food Law regulation to traceability of meat products (EC 2002). Therefore, different business collaboration process models apply to farmers and to meat processors. In addition, both sets of regulations contradict with business collaboration process model given in Figure 3-2. Second, some large food operators have already expensive legacy transparency systems in place that are not (fully) compatible with the reference architecture and the associated reference data query and data capture process models. Third, many other supply chain actors do not have the resources to deploy the required IT systems (*i.e.* the repository and application software) (Kassahun *et al.* 2014). As a result, the business collaboration processes and the distributed private transparency systems used in meat supply chains do not comply with any one reference architecture, are misaligned, and fall short of providing chain-wide transparency.

The misalignment encountered in the case study and in many other multi-organisational collaborations can be classified into three types. The first type of misalignment occurs when the business collaboration processes adopted in the collaboration network are incompatible with one another, *i.e.* business process to business process (BP2BP) misalignment. For instance, the regulations that apply to farmers and slaughterhouses are in some respects different from regulations that apply to slaughterhouses and other meat processors leading to different business collaboration processes in the different segments of the collaboration network. The second type of misalignment occurs between the business collaboration processes (assuming they are aligned) and the distributed IT system (again assuming there is integrated distributed IT system), *i.e.* business process to IT (BP2IT) misalignment. This occurs, for example, when some activities of the chosen business collaboration process are not supported by the underlying distributed IT system; or, the possibilities presented by the distributed IT system are not exploited by the business collaboration process. Finally, the third type of misalignment occurs when the underlying IT systems of the collaborating organisations are not aligned and thus cannot interoperate, *i.e.* IT system to IT system (IT2IT) misalignment. The three alignment concerns are depicted in Figure 3-3.

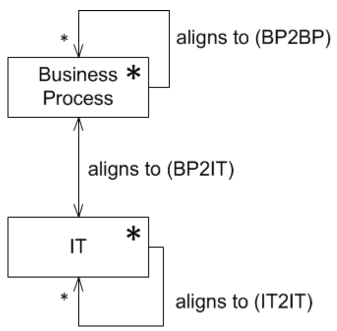


Figure 3-3: Business-IT alignment problem in multi-organisational collaboration

To address these problems, we need to provide design abstractions for explicitly depicting the BP2BP, BP2IT, and IT2IT misalignments. These design abstractions together with the corresponding systematic approach will be discussed in the following section.

3.4 BITA* Framework

BITA* stands for Business process-IT Alignment (BITA) framework for multiple collaborating organisations—the symbol '*' denotes that multiple organisations are involved. A design framework consists of multiple design viewpoints. A design *viewpoint* provides a template for designs with respect to specific concerns of specific group of stakeholders. A design that follows the conventions, including models and notations, of a particular viewpoint is referred to as a *view*. The notion of design viewpoint is in particular applied in the context of software architecture (ISO/IEC/IEEE 2011). Example architecture viewpoints include decomposition viewpoint for modelling the partitioning of software code into modules and sub modules, and deployment viewpoint for modelling the assignment of executable components to hardware nodes (Clements *et al.* 2010).

Designing a framework requires defining the required modelling abstractions with the help of a metamodel and defining the corresponding systematic approach for applying the modelling abstractions. The following sub-sections present the elements of the BITA* framework as follows. Section 3.4.1 presents the BITA* metamodel, section 3.4.2 describes the systematic approach, section 3.4.3 presents the alignment viewpoints and the associated allocation and alignment modelling abstractions, and finally, section 3.4.4 presents relational models for storing and managing allocation and alignment matrices.

3.4.1 Metamodel

The elements of a design framework are described by its metamodel. The BITA* framework metamodel is depicted in Figure 3-4. In the following the metamodel is described using the concepts and terminologies of the ISO/IEC/IEEE standard.

BITA* consists of three viewpoints, which are *BP2BP*, *BP2IT* and *IT2IT* viewpoints, corresponding to the three alignment categories. Two types of stakeholders are identified: *business analysts* and *software architects*. In addition we identify the *interdisciplinary teams* of business analysts and software architects also as a stakeholder group type. The BP2BP viewpoint provides the allocation and alignment models for business analysts. The BP2IT viewpoint provides the allocation and alignment models for interdisciplinary teams of business analysts and software architects. The IT2IT alignment viewpoint provides the allocation and alignment models for software architects.

BITA* contains *allocation* and *alignment* model types. Each of the three viewpoints contains specific allocation and alignment model types that are applicable to the stakeholders of the viewpoint. This means that the BP2BP viewpoint contains BP2BP allocation and alignment models, the BP2IT viewpoint contains BP2IT allocation and alignment models, and the IT2IT viewpoint contains IT2IT allocation and alignment models.

As stated earlier, allocation and alignment modelling abstractions are the core of BITA* and we have explained informally what alignment entails. We have stated that allocation models represent business collaboration models. In the metamodel we identify the three business collaboration model types that help define alignment explicitly, which are BCP, DIT and BCP/DIT. A Business Collaboration Process (BCP) is represented using a BPMN collaboration model. The distributed IT (DIT) and its relationship with the business collaboration process models (BCP/DIT) are represented using SOA IT service models. BPMN is a graphical model. SOA IT service models include many separate models, prominently IT service descriptions (W3C 2007), message (data) exchange protocols (Mitra 2003, Bouguettaya *et al.* 2014) and IT service discovery (OASIS 2004, Crasso *et al.* 2013).

In general, BCP models are not comparable with each other; likewise the DIT and BCP/DIT models. To make them comparable we convert them into matrix-based models. The term allocation refers to the transformation of BCM, DIT and BCM/DIT models into matrix forms when necessary with the help of workflow patterns. We distinguish two relevant workflow patterns: Control Flow Patterns (CFPs) and DataFlow Patterns (DFPs).

To convert BCP models into allocation matrices we use CFPs. To convert DIT models into allocation matrices we use DFPs. The conversion of BCP/DIT models to matrices does not require workflow patterns. It is hereby important to realize that we use limited aspects of DIT modelling. A complete DIT model may require the use of architectural patterns for distributed computing (Buschmann *et al.* 2007).

We introduce the concepts *reference* and *concrete* to distinguish between models and allocation matrices derived from the *reference architecture* from models and allocation matrices derived from the architectures adopted by the individual organisations (referred to as *concrete architectures*). Ideally alignment is modelled by comparing concrete allocation matrices with the corresponding reference allocation matrix. However, in practice, the required concrete allocation matrices are often unavailable. Fortunately, organisations can use the reference allocation metrics to indicate how their view of business collaboration matches, or mismatches, with the reference architecture.

To specify the match or mismatch between allocation matrices we introduce the concept of *alignment attribute*. Three basic alignment attributes, borrowed from the reflexion modelling approach (Murphy *et al.* 2001), are: *convergence, divergence* and *absence.*

3.4.2 Systematic Approach

In the previous section we stated that alignment modelling compares two sets of business collaboration models. The ideal approach to alignment is based on comparing concrete business collaboration models with the corresponding reference collaboration model. This means that there will be as many comparisons as there are organisations for a given reference collaboration model. We also indicated that such an ideal situation, in which organisations produce the required quality business collaboration models, rarely exist.

In the following, we first describe the alignment process assuming the ideal scenario in which that required concrete allocations can be obtained. We then provide an alternative step-by-step guidance on how to produce alignment matrices based solely on reference models and informal descriptions of the concrete models.

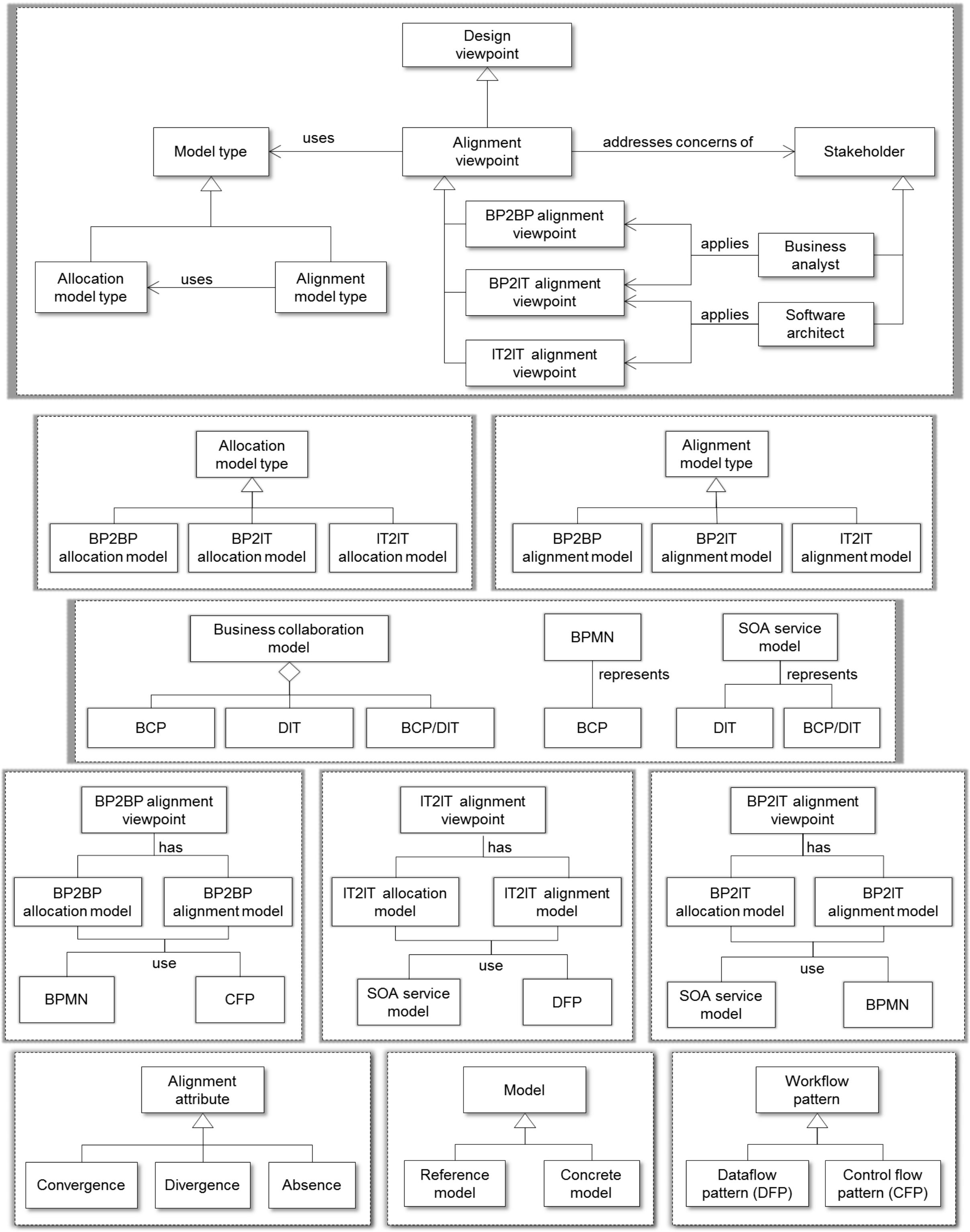


Figure 3-4: The BITA metamodel*

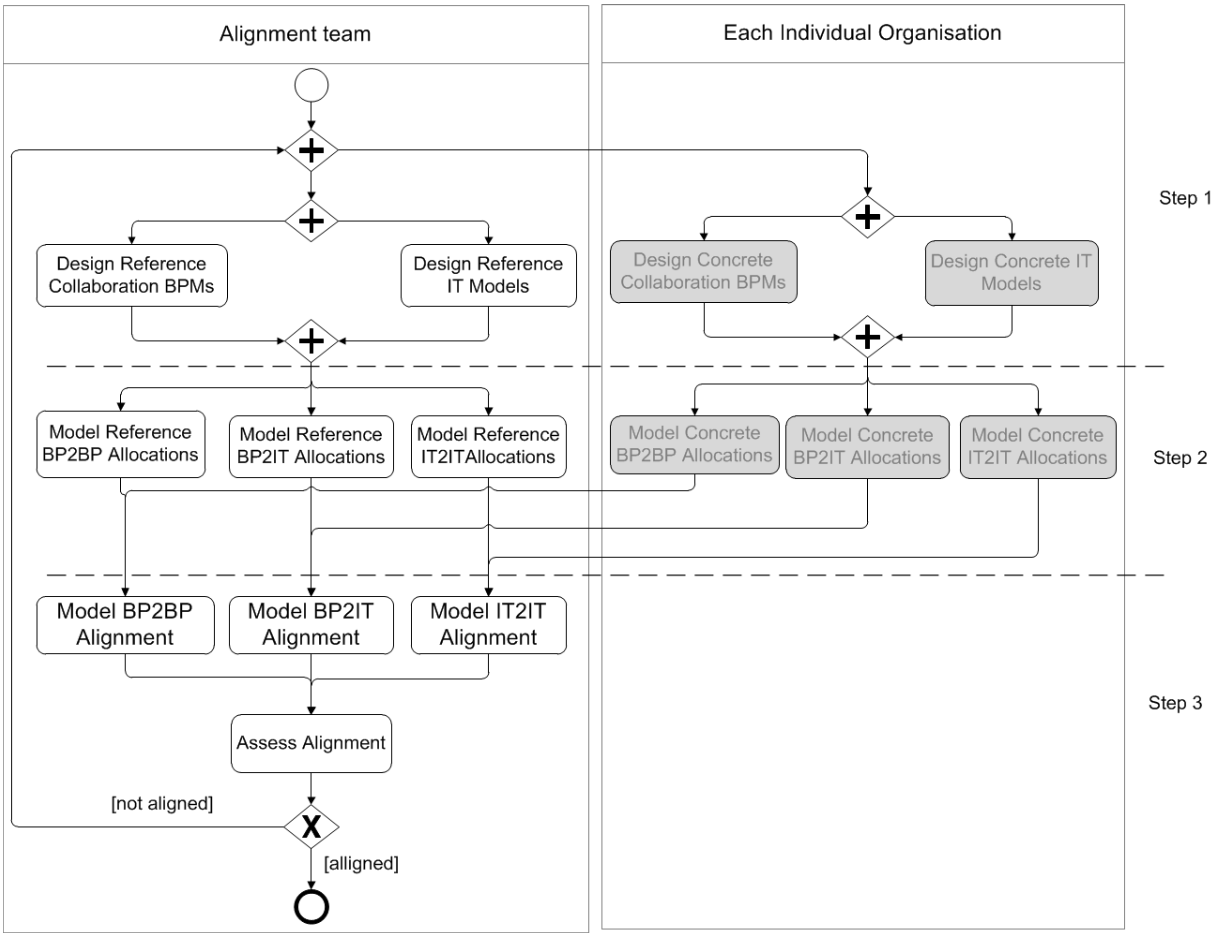


Figure 3-5: The BITA alignment process*

The overall approach used in BITA* is depicted in Figure 3-5. The steps for concrete models and allocations are shaded to indicate those steps are optional. The approach consists of three basic steps, which we describe below:

1. Design the Required Models

In this step the reference and concrete business collaboration models are designed. The business collaboration processes are modelled using BPMN and control-flow patterns. Reference and concrete collaboration modelling are performed in parallel.

2. Model the Allocations

In this step the BP2BP, BP2IT and IT2IT reference and concrete allocation matrices are derived. Reference allocation matrices are derived from reference models. Concrete allocation matrices are derived from concrete models. For each reference matrix there may be as many concrete matrices as there are organisations. The allocation matrices are derived using the allocation models provided in section 3.4.3. It suffices to state here that an allocation matrix is a two dimensional table in which the cells are assigned a binary (*true*/*false*) value. In this paper we use a *tick mark* to indicate *true* (allocated) and an *x* or *blank cell* to represent *false* (not allocated).

3. Model the Alignments

In this step a reference allocation matrix is compared with the corresponding concrete allocation matrices. The alignment matrix is a copy of the corresponding reference allocation matrix extended with all elements that are only found in the corresponding concrete allocation matrices. The results of the comparison are indicated by alignment attributes, which are *convergence, absence* or *divergence*. The attributes are now defined explicitly as follows:

Convergence: Consider a cell in a reference allocation matrix that is assigned *true* (allocated). If the corresponding cell in every concrete allocation matrix is also assigned *true*, then we assign the corresponding cell of the alignment matrix as *convergent*.

Absence: Consider a cell in a reference allocation matrix that is assigned *true* (allocated). If the corresponding cell in every concrete allocation matrix is assigned *false*, then we assign the corresponding cell of the alignment matrix as *absent*.

Divergence: Consider all corresponding cells in all corresponding concrete allocation matrices that are assigned *true* (allocated). If the corresponding cell in the corresponding reference allocation matrix is missing or assigned *false*, then we assign the corresponding cell of the alignment matrix as *divergent*.

The three attributes are sufficient if organisations are compared pairwise. However, a reference matrix is compared with as many concrete matrices as there are organisations. This will lead to *partial convergence, partial absence* or *partial divergence*. We define these attributes as follows:

Partial convergence: Consider a cell in a reference allocation matrix that is assigned *true* (allocated). If the corresponding cells of some (but not all) of concrete allocation matrices are also assigned *true*, then we assign the corresponding cell of the alignment matrix as *partially convergent*.

Partial absence = Partial convergence

Partial divergence: Consider a cell in any of the concrete allocation matrices that is assigned *true* (allocated). If the corresponding cell in the corresponding reference allocation matrix is missing or assigned *false*, then we assign the corresponding cell of the alignment matrix as *partial divergent* if it is not already assigned *divergent*.

For completeness, we also include an alignment attribute called *invalid* to indicate that the allocation is invalid or impossible in both the reference and the concrete matrices. The alignment attributes are summarized in Table 3-1.

Now that we have covered the ideal scenario in which the concrete models and the corresponding allocation matrices are available, below we consider the alternative approach. The alternative approach addresses the problem that explicit concrete models and the corresponding allocation matrices are not available, or not available in the required quality. The alternative approach circumvents the problem of missing explicit concrete models by deriving alignment matrices from the reference allocation matrices and informal descriptions of the concrete models.

Table 3-1: Alignment attributes

Alignments	*Allocations*		*Notations*
	reference	*Concrete*	
Convergence	√	√	+
Absence	√	*x*	–
Divergence	*x*	√	~
Partial convergence	√	√ | x	±
Partial divergence	x	√ | x	#
Invalid	*x*	*x*	*x (or left empty)*

The steps below describe how a single alignment (a single cell in an alignment matrix) can be filled in with alignment attribute based on a round table discussion with representatives of the collaborating organisations:

1. Present (or describe) the reference allocation to the representatives of the collaborating organisations and ask for their view about it.
2. If all of them agree with the reference allocation, then fill in *convergence* (+) in the corresponding cell of the alignment matrix.
3. If some of them agree, while others not, with the reference allocation, then fill in *partial convergence* (±) in the corresponding cell of the alignment matrix.
4. If all of them disagree with the reference allocation, then fill in *absence* (–) in the corresponding cell of the alignment matrix.
5. If all of them come up with an alternative allocation, include the cell in the alignment matrix, then fill in *divergence* (~) in the new cell of the alignment matrix.
6. If some (not all) of them come with an alternative allocation, include the cell in the alignment matrix, then fill in *partial divergence* (#) in the new cell of the alignment matrix.

3.4.3 Alignment Viewpoints

In this section, we present the three viewpoints of BITA* and the corresponding modelling abstractions. In section 3.4.3.1 we present the BP2BP alignment viewpoint, in section 3.4.3.2 we present the BP2IT viewpoint, and in section 3.4.3.3 we present the IT2IT viewpoint.

3.4.3.1 BP2BP Alignment Viewpoint

The BP2BP alignment viewpoint provides a BP2BP allocation model and a BP2BP alignment model. The BP2BP allocation model is used for representing business collaboration processes in a matrix form. The BP2BP alignment model is used for comparing two corresponding BP2BP allocation matrices with each other. The allocation model uses CFPs to convert BPMN models to BP2BP allocation matrices.

Before we define the allocation and alignment models we describe CFPs. Several CFPs have been defined and categorized by van der Aalst and ter Hofstede which are summarized in Table 3-2. We identify four categories of CFPs: *branch and sync*, *iteration*, *multiple instance* and *event-driven*. *Branch and sync* CFPs define the sequencing of activities, such as linear (sequential), branching and parallel. *Iteration* CFPs define how the same sets of activities are performed repetitively. Iteration can be a loop or a recursion. *Multiple instance* CFPs define how the same sequence of activities is executed in parallel in separate threads of execution.

Event driven CFPs define the effects of expected and unexpected events, such as starting, cancelling and completion. CFPs can be arranged hierarchically (*i.e.* a CFP is composed of other CFPs, activities or both) and as such are a powerful means of capturing business process models at different levels of detail.

Table 3-2: Control-flow patterns (adapted from, van der Aalst and ter Hofstede 2011)

Pattern Categories	Patterns*
Branch and Sync	Sequence (1), Parallel Split (2), Synchronization (3), Exclusive Choice (4), Simple Merge (5), Multi-Choice (6), Structured Synchronizing Merge (7), Multi-Merge (8), Structured Discriminator (9), Blocking Discriminator (28), Cancelling Discriminator (29), Structured Partial Join (30), Blocking Partial Join (31), Cancelling Partial Join (32), Generalized AND-Join (33), Local Synchronizing Merge (37), General Synchronizing Merge (38), Thread Merge (41), Thread Split (42), Deferred Choice (16), Interleaved Parallel Routing (17), Milestone (18), Critical Section (39), Interleaved Routing (40)
Iteration	Arbitrary Cycles (10), Structured Loop (21), Recursion (22)
Multiple Instance	Multiple Instances without Synchronization (12), Multiple Instances with a Priori Design-Time Knowledge (13), Multiple Instances with a Priori Run-Time Knowledge (14), Multiple Instances without a Priori Run-Time Knowledge (15), Static Partial Join for Multiple Instances (34), Cancelling Partial Join for Multiple Instances (35), Dynamic Partial Join for Multiple Instances (36)
Event-driven	Transient Trigger (23), Persistent Trigger (24), Cancel Task (19), Cancel Case (20), Cancel Region (25), Cancel Multiple Instance Activity (26), Complete Multiple Instance Activity (27), Implicit Termination (11), Explicit Termination (43)

* The pattern names used by the authors are shortened for the sake of readability; the pattern IDs (given insides brackets) are, however, original.

3.4.3.1.1 BP2BP Allocation Model

The BP2BP allocation model represents a business collaboration process model (such as the one shown in Figure 3-2) as matrices. Activities, control flows and organisations are key elements of business collaboration process models. A business collaboration process model is essentially a specification of which organisation is responsible for which activity and how the control ‘flows’ from one activity to the next.

The BP2BP allocation model consists of two types of allocations: *activity allocations* and *CFP allocations*. *Activity allocations* are derived directly from a business collaboration process model by identifying the pool (and thus the organisation) the activity belongs to. An *activity allocation matrix* is a collection of activity allocations of a business collaboration

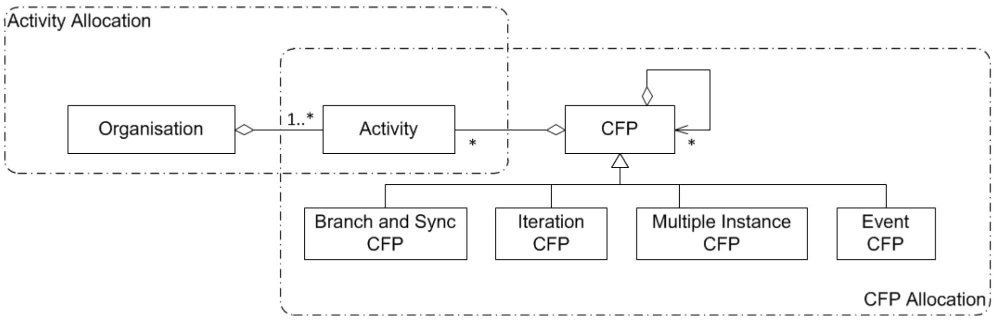


Figure 3-6. BP2BP allocation model

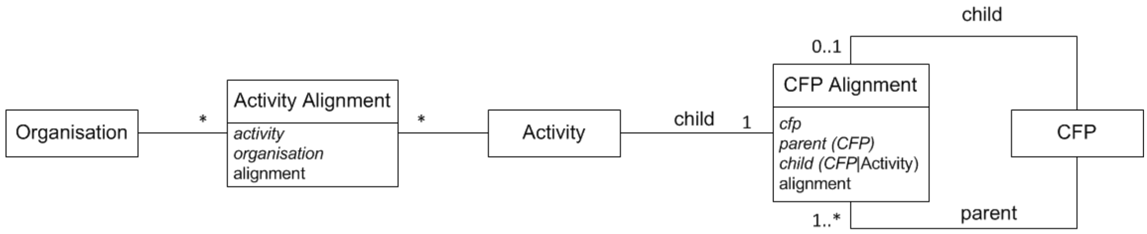


Figure 3-7: The BP2BP alignment model

process model represented in a matrix form. To make *CFP allocations* first all CFPs must be identified. Then, both activities and CFPs are allocated to a CFP. The allocation of CFPs is better described with an example, which we provide later (in Figure 3-13 and Table 3-7). A *CFP allocation matrix* is a collection of CFP allocations of a business collaboration process model represented in a matrix form. Figure 3-6 depicts the BP2BP allocation model.

3.4.3.1.2 BP2BP Alignment Model

The BP2BP alignment model consists of *activity* and *CFP alignment* matrices (see Figure 3-7), corresponding to the activity and CFP allocation matrices, respectively.

Activity alignment matrix is the result of comparing reference activity allocation matrix with the corresponding concrete activity allocation matrices. An activity alignment matrix is a two dimensional matrix whose axes are *activity* and *organisation*. (The cells of all alignment matrices are assigned one of the five *alignment attributes*.) *CFP alignment matrix* is the result of comparing reference CFP allocation matrix with the corresponding concrete CFP allocation matrices. A CFP alignment matrix is a three dimensional matrix whose axes are *CFP*, *Parent (CFP)* and *Child (CFP* or *Activity).*

3.4.3.2 BP2IT Alignment Viewpoint

The BP2IT alignment viewpoint provides a BP2IT allocation model and a BP2IT alignment model. The BP2IT allocation model is used for representing the relationships between business collaboration process models and the supporting distributed IT in a matrix form. The BP2IT alignment model is used for comparing two corresponding BP2IT allocation matrices with each other.

3.4.3.2.1 BP2IT Allocation Model

The BP2IT allocation model represents how the business processes (PMs) of a business collaboration process model (such as those shown in Figure 3-2) are supported by IT services (such as those defined in EPCIS) as matrices. Activities, data objects, PMs and IT services are key elements of BP2IT allocation.

The BP2IT allocation model consists of two types of allocations: *IT service allocations* and *(data input/data output) I/O allocations. IT service allocations* describe the relationships among *IT services*, *clients (i.e.* the *activities* that use the IT services, and by extension also the organisations that perform the activities) and *providers (i.e.* the *PMs* that realize the IT services, and by extension also the organisations that support the IT services). An *IT service allocation matrix* is a collection of IT service allocations represented in a matrix form. *I/O allocations* to an activity describe the data inputs to the activity and the data outputs from the activity. An I/O allocation matrix can actually be split into data input object allocation matrix

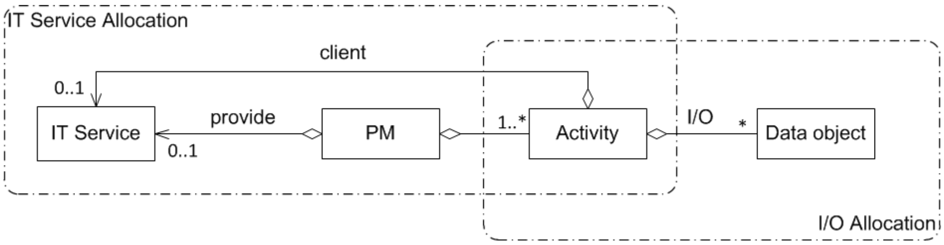


Figure 3-8: BP2IT allocation model

and data output object allocation matrix. An *I/O allocation matrix* is a collection of all I/O allocations represented in a matrix form. Figure 3-8 depicts the BP2IT allocations model.

3.4.3.2.2 BP2IT Alignment Model

The BP2IT alignment model consists of *IT service* and *I/O alignment* matrices (see Figure 3-9), corresponding to the service and I/O allocation matrices, respectively.

IT service alignment matrix is the result of comparing reference IT service allocation matrix with the corresponding concrete IT service allocation matrices. An IT service alignment matrix is a three dimensional matrix whose axes are *IT services*, *clients* (*activities*) and *providers* (*PMs*). *I/O alignment matrix* is the result of comparing reference I/O allocation matrix with the corresponding concrete I/O allocation matrices. An I/O alignment matrix is a three dimensional matrix whose axes are *activity*, *data object* and *organisation*. The I/O alignment matrix can be divided into two separate input data object and output data object alignment matrices.

3.4.3.3 IT2IT Alignment Viewpoint

The IT2IT alignment viewpoint provides an IT2IT allocation model and an IT2IT alignment model. The IT2IT allocation model is used for representing a distributed IT system. A distributed IT system is modelled as a set of IT services and data sharing (message exchanges) among the IT services. The IT2IT alignment model is used for comparing two corresponding IT2IT allocation models with each other. The allocation model uses DFPs to represent data sharing as allocation matrices.

Before we define the IT2IT allocation and alignment models we describe briefly which DFPs we will use. DFPs are used to capture well-known data flow patterns. Table 3-3 lists the DFPs that are relevant for representing data sharing concerns in multi-organisational collaboration

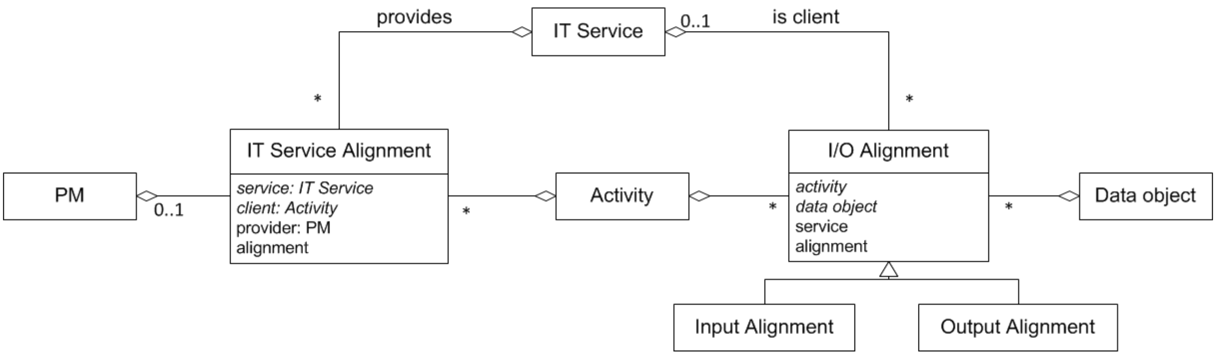


Figure 3-9: BP2IT alignment model

context. The patterns are categorized into four categories by van der Aalst and ter Hofstede, namely: *visibility*, *interaction*, *transfer* and *routing* DFPs. This categorisation is important because the data access and usage concerns fall also into these categories. *Visibility* DFPs define the scope of accessibility of a data object. For instance, an *activity*[2] *scope* signifies that the data object visibility is restricted to the activity instance; while an *instance scope* signifies that its visibility extends to all activities of a business process instance. *Interaction* DFPs define how the data object visibility changes due to interaction. For instance, *activity to activity* means that the data object remains in an activity scope during interaction; while *to multiple instance activity* means that the interaction changes from activity scope to *multiple instance* scope. *Transfer* DFPs define the mechanisms of data interaction, which can be by value, by reference, etc. *Routing* DFPs define how a data object affects the control flow, such as launching or ending an activity, or altering the flow of control.

Table 3-3: Workflow data patterns (adapted from, van der Aalst and ter Hofstede 2011)

Categories	Patterns*
Visibility	Activity (1), Multiple Instance (4), BP Instance (5), External (8)
Interaction	
Internal	Activity to Activity (9), To Multiple Instance Activity (12), From Multiple Instance Activity (13), Instance to Instance (14)
External	Activity pushes data (15), Activity pulls data (16), Data are pushed to Activity (17), Activity receives data (18), BP Instance pushes data (19), Data are pulled from BP Instance (20), Data are pushed to BP Instance (21), BP Instance pulls data (22)
Transfer	Incoming By Value (27), Outgoing by Value (28), Copy In/Copy Out (29), By Unlocked Reference (30), By Locked Reference (31), Input Transformation (32), Output Transformation (33)
Routing	Existence as Activity Precondition (34), Value as Activity Precondition (35), Existence as Activity Postcondition (36), Value as Activity Postcondition (37), Event-based Activity Trigger (38), Data-based Activity Trigger (39), Data-based Routing (40)

* The pattern names used by the authors are shorted for the sake of readability; the pattern IDs (given insides brackets) are, however, original. DFPs deemed irrelevant for the purpose of this paper are not included.

3.4.3.3.1 IT2IT Allocation Model

The IT2IT allocation model represents the distributed IT system. This is partially done in the BP2IT viewpoint where IT services are associated with PMs (and thus organisations). The remaining aspects of the distributed IT are described by IT systems, IT services, organisations, data objects and DFPs.

The IT2IT allocation model consists of three types of allocations: *IT system allocations*, *data object allocations* and *DFP allocations*. *IT system allocations* describe which organisation provides which IT systems and how the IT services are distributed among the IT systems. An *IT system allocation matrix* is a collection of IT system allocations of the distributed IT represented in a matrix form. *Data object allocations* describe which organisations provide which data objects. A *data object allocation matrix* is a collection of data object allocations in

[2] We use the term *activity* instead of *task* (the term originally used in DFPs) in order to be consistent with the terminology of BPMN. We also use the term *activity* instead of *task* and *block task; instance* instead of *case*; *business process* instead of *workflow*, and (*external*) *data store* instead of *environment.*

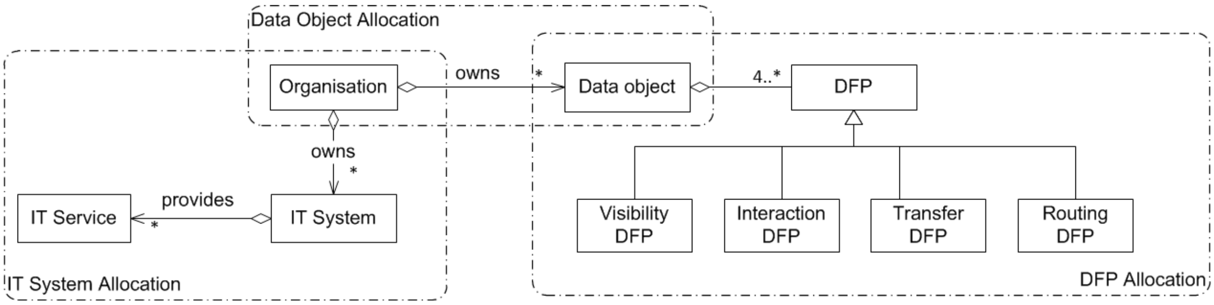


Figure 3-10: IT2IT allocation model

the collaboration network represented in a matrix form. *DFP allocations* describe how data objects are shared and used. A *DFP allocation matrix* is a collection of DFP allocations for all data objects represented in a matrix form. A data object can be assigned up to four DFPs corresponding to the four data access and usage concerns. Therefore, a DFP allocation matrix can, in fact, be split in to four DFP allocation matrices. Figure 3-10 depicts the IT2IT allocations model.

3.4.3.3.2 IT2IT Alignment Model

The *IT2IT* alignment model consists of *IT system, data object* and *DFP alignments* matrices (see Figure 3-11), corresponding to the IT system, data object and DFP allocation matrices, respectively.

IT system alignment matrix is the result of comparing reference system allocation matrix with the corresponding concrete system allocation matrices. An IT system alignment matrix is a two dimensional matrix whose axes are *IT systems* and *IT services*. *Data object alignment matrix* is the result of comparing reference data object allocation matrix with the corresponding concrete data object allocation matrices. A data object alignment matrix is a two dimensional matrix whose axes are *data objects* and *organisations*. *DFP alignment matrix* is the result of comparing reference DFP allocation matrix with the corresponding concrete DFP allocation matrices. A DFP alignment matrix is a three dimensional matrix whose axes are *data object*, *dfp* and *organisation*. Since a data object is potentially associated to four DFPs, the DFP alignment matrix can be split into four separate DFP alignment matrices.

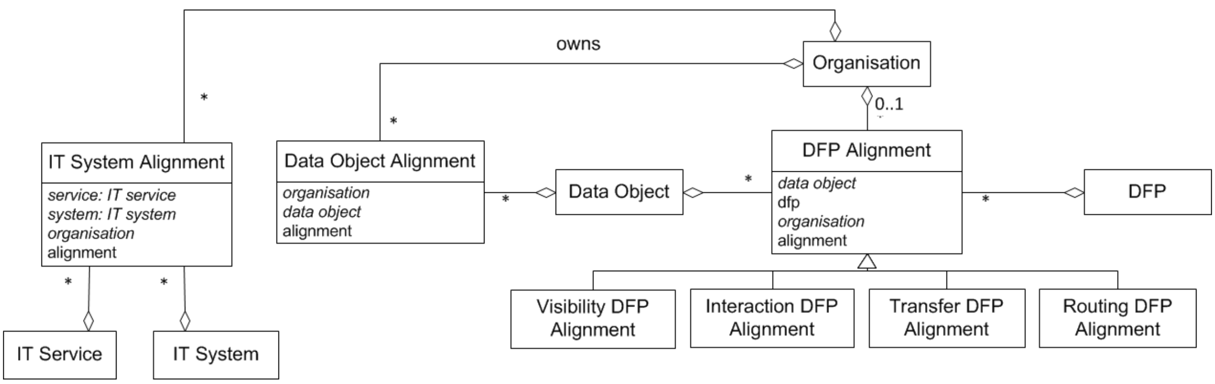


Figure 3-11: IT2IT alignment model

3.4.4 Relational Model

In the previous section we have provided the alignment viewpoints and the corresponding allocation and alignment models. In this section we provide relational models for representing the models provided in Figure 3-6 through Figure 3-11 in relational databases. Table 3-4 provides the relational schema for the models. The schema includes only the essential columns of the tables. Thus, for instance, an activity allocations matrix may contain the names of the activities and organisations, but these columns are ignored in the schema since they are not essential aspects of de schema modelling.

Table 3-4: Relational schema for allocation and alignment models

Allocation and alignment models	*Relational schema*
BP2BP Viewpoint	
Allocation model	
Activity allocations	ActAlloc (act-id, org-id, [alloc])
CFP allocations	
CFP structure	CFPStrAlloc (cfp-id, pcfp-id, [alloc])
CFP activities	CFPActAlloc (cfp-id, act-id, [alloc])
Alignment model	
Activity alignments	ActAlign (act-id, org-id, align-attr)
CFP alignments	
CFP structure	CFPStrAlign (p-id, pcfp-id, align-attr)
CFP activities	CFPActAlign (cfp-id, act-id, align-attr)
BP2IT Viewpoint	
Allocation model	
IT-service allocation	SrvAlloc (srv-id, pm-id, org-id, [alloc])
I/O allocation	
Inputs allocations	InAlloc (act-id, do-id, [alloc])
Output allocations	OutAlloc (act-id, do-id, [alloc])
Alignment model	
IT-service allocation	SrvAlign (srv-id, pm-id, org-id, [alloc])
I/O allocation	
Inputs allocations	InAlign (act-id, do-id, align-attr)
Output allocations	OutAlign (act-id, do-id, align-attr)
IT2IT Viewpoint	
Allocation model	
System allocation	SysAlloc (sys-id, srv-id, org-id, [alloc])
Data object allocation	DOAlloc (do-id, org-id, [alloc])
DFP allocations	DFPAlloc (do-id, dfp-id, [org-id], [alloc])
Alignment model	
System alignment	SysAlign (sys-id, srv-id, org-id, align-attr)
Data object alignments	DOAlign (do-id, org-id, align-attr)
DFP alignments	DFPAlign (do-id, dfp-id, [org-id], align-attr)

In Table 3-5 we provide the SQL queries to determine the alignment attributes using relational algebra notation. R represents a particular reference allocation matrix represented as a relational table; likewise C represents its concrete counterparts. The attributes *id-1*, *id-2 … id-n* represent the columns of the primary key that will be used to select a particular allocation for comparison from reference allocation matrix (R) and its concrete (C) counterparts. For instance, for a given reference ActAlign allocation matrix and its concrete counterparts, the two primary key attributes that identify a particular activity allocation are *act-id* and *org-id*. The attribute *alloc* will generally not be required because a query using an *act-id* and *org-id* will return a single row, indicating *allocated* (*true*), or now rows, indicating *not allocated* (*false*). However, the attribute *alloc* is included for reasons of understandably. For a given allocation R-count represents the number of allocations in a reference matrix, which should be 0 or 1. C-count represents the number of allocations in the corresponding concrete matrices, which should be between 0 and n, where *n* is the number of organisations.

Table 3-5: SQL query for retrieving alignment attributes

Alignment attribute	*Query*
R-count (*id-1*, *id-2*, … *id-n*)	σ (count(R.id-1 = id-1 ∧ R.id-2 = id-2 ∧ … ∧ R.id-n = id-n ∧ R.alloc=YES))
C-count (*id-1*, *id-2*, … *id-n*)	Σ (σ (count(C.id-1 = id-1 ∧ C.id-2 = id-2 ∧ … ∧ C.id-n = id-n ∧ C.alloc=YES)))
covergence	R-count = 1 ∧ C-count = n
partial covergence	R-count = 1 ∧ 0 < C-count < n
divergence	R-count = 0 ∧ C-count = n
partial divergence	R-count = 0 ∧ 0 < C-count < n
absence	R-count = 1 ∧ C-count = 0

3.5 APPLYING BITA* TO THE CASE STUDY

We now apply the BITA* approach to the case study described in section 3.3. We first provide more information about the reference and concrete models in the following sub section. We then provide the views, which contain the alignment models, that correspond to the three viewpoints of the BITA* framework in section 3.5.2.

3.5.1 More on the Generic Reference Architecture

We have already defined a reference query business collaboration process model in Figure 3-2 that was derived from the EPCIS reference architecture. Next we describe the reference SOA and data models that support the process collaboration process model.

The EPCIS reference architecture specifies two basic IT services: data capture IT service (CaptureSrv) and data query IT service (QuerySrv). The CaptureSrv service corresponds to the data capture business process. Data capturing is a local business process; therefore, CaptureSrv service of one organisation is not accessible to its collaboration partners. There are two types of query services, $QuerySrv_{ITS}$ and $QuerySrv_{ETS}$, that correspond to PM_{ITS} and PM_{ETS}, respectively. The reference architecture does not describe how the $QuerySrv_{ETS}$ should be composed from distributed $QuerySrv_{ITS}$ services. The process models PM_{ETS} depicted in Figure 6-2 is just one way of realizing $QuerySrv_{ETS}$. The $QuerySrv_{discovery}$ service corresponds to the $PM_{discovery}$ process. $QuerySrv_{app}$ realizes the PM_{app} process model.

The services fulfil one or two of the SOA roles. $QuerySrv_{ETS}$ is a *client* of $QuerySrv_{discovery}$ and $QuerySrv_{ITS}$ services. In turn, $QuerySrv_{app}$ is a *client* of $QuerySrv_{ETS}$. In both cases, the latter are said to be *provider* of a service to the former. At any one time, a food operator either provides either $QuerySrv_{ETS}$ or $QuerySrv_{ITS}$. Third parties are, according to the reference architecture, *service brokers*.

The reference architecture defines also a data model for transparency systems, which is called the EPCIS event model. An *event* data object contains four data objects called event dimensions. They are conveniently called the *what*, the *when*, the *where* and the *why* of events. The *what* data object is an ID and represents the unique identification of a product item the event is about. The *when* data object is a time stamp and represents the data and time the event occurred. The *where* data object is an ID and represents the place where the event occurred. And, the *why* data object is a predefined vocabulary and represents the reason for recording the event. IDs and predefined vocabularies are largely meaningless to human readers. The descriptive information corresponding to IDs and vocabularies are retrieved from master data repositories (GS1 2014) by accessing applications. Yet another data object is a service URL (*srvURL*) that identifies the web address of a QuerySrv$_{ITS}$. This data object is used by discovery IT service.

3.5.2 Modelling Alignments

Now that we have the reference business collaboration models; *i.e.* the reference business collaboration process model shown in Figure 3-2, the business collaboration IT model described in section 3.3, and a model for the relationship between these two business collaboration models described in section 3.5.1; we can proceed to demonstrate alignment modelling using the case study as a running example. The alignment modelling is based the reference business collaboration models just described and informal descriptions of the concrete business collaboration models.

To illustrate how informal descriptions of a concrete business collaboration models look like, we provide an example. Figure 3-12 is an example of informal description of a concrete architecture for a chain-wide transparency system provided by a meat processor during the case study. It represents just one of the many views on how the different organisations view a chain-wide transparency system. The architecture prescribes a business collaboration model that is significantly different from the reference architecture. In the reference architecture the data capturing is a local process; data querying is a collaborative process. In this concrete architecture, however, the meat operator opens its data capture IT services to be used by its partners, making data capturing a collaboration process. Besides, a number new IT services involving third parties, such as QS (a quality assurance agency) and HIT (a national bovine animal registration office) are introduced. For the detailed description of similarities and differences between the concrete business collaboration models and the EPCIS-based reference business collaboration models refer to Kassahun *et al.* (2014).

In the following sections we apply the alternative alignment approach described in section

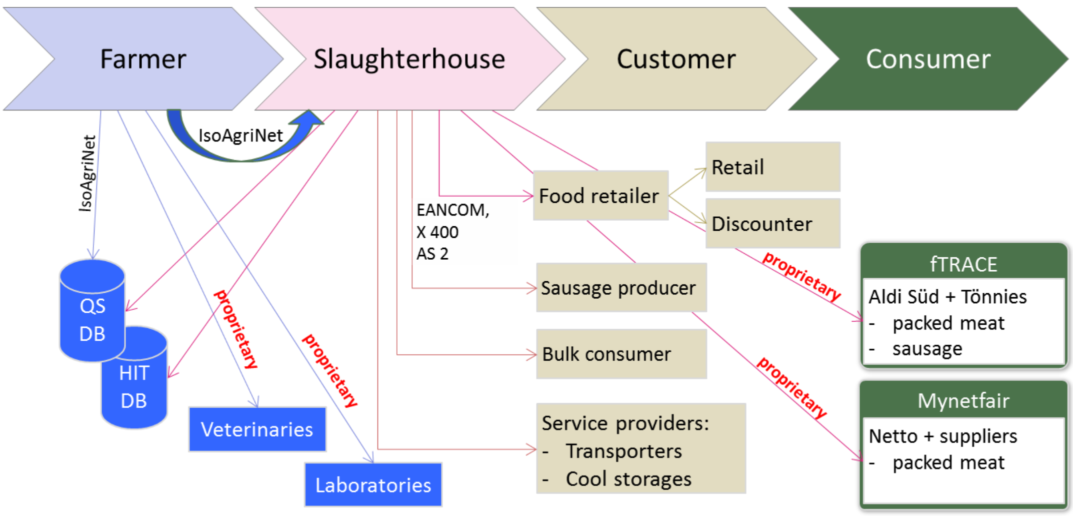


Figure 3-12: Concrete architecture of transparency systems at a meat processor (Kassahun et al. 2014)

3.4.2. The alternative approach to alignment applies to cases in which the concrete models are only available as informal descriptions and, therefore, no concrete allocation matrices are produced. We generate the reference allocation matrices from the reference models and then use the reference allocation matrices and the informal descriptions of the corresponding concrete models to derive the alignment matrices. The alignment modelling is organised in three views corresponding to the three viewpoints of the BITA* framework.

3.5.2.1 BP2BP View

In this section we present the BP2BP allocation matrices derived from the reference business process model depicted in Figure 3-2. We then present the BP2BP alignment matrices that are derived from the reference BP2BP allocation matrices and informal descriptions of the concrete business collaboration process models.

3.5.2.1.1 BP2BP Allocations

Table 3-6 shows the reference activity allocations following the business process model presented in section 3.3. Deriving activity allocation is straightforward—activities are listed as rows and organisations (pools) are listed as columns. The columns and rows can directly be read from the collaboration model (Figure 3-2) or the choreography model (Figure 3-13).

Table 3-6: Reference activity allocations

Activities	*Organisations*	
	Food operators	*Third party*
a1: end-user query		√
a2: decide where to query	√	
a3: local query	√	
a4: lookup		√
a5: iterative remote query	√	
a6: recursive query over ingredients	√	
a7: visualize		√

To model the CFP allocations the CFPs have to be identified from the collaboration model given by Figure 3-2. Figure 3-13 shows a choreography model based on the collaboration model given by Figure 3-2. The CFPs are depicted as overlapping blocks (dashed lines).

Table 3-7 depicts the reference CFP allocation matrix. (A CFP allocation matrix is a three dimensional matrix represented as a two dimensional table. We will also hereafter rollup all multidimensional matrices into two dimensional tables.). Pattern *p1* is a *sequence* (*cfp-1*) CFP since the three patterns (*p2, p3, p4*) and the two activities (*a1*, *a7*) are arranged sequentially. Patterns *p2* is a *transient trigger* (*cfp-23*) event CFP; *p3* is an *implicit termination* (*cfp-11*) event CFP. Pattern *p4* is a *recursion* (*cfp-20*) CFP since activity *a6* recursively triggers its containing CFP pattern *p4*. Pattern *p5* is a multi-choice (*cfp-6*) CFP since one or both of the two parallel paths can be executed. The parallel paths merge in pattern *p6*, which is multi-merge (*cfp-8*) CFP. Pattern *p7* is a sequence (*cfp-1*) CFP since the lookup activity (*a4*) and pattern *p8* are arranged sequentially. Pattern *p8* is a s*tructured loop* (*cfp-21*) CFP since remote queries are initiated iteratively.

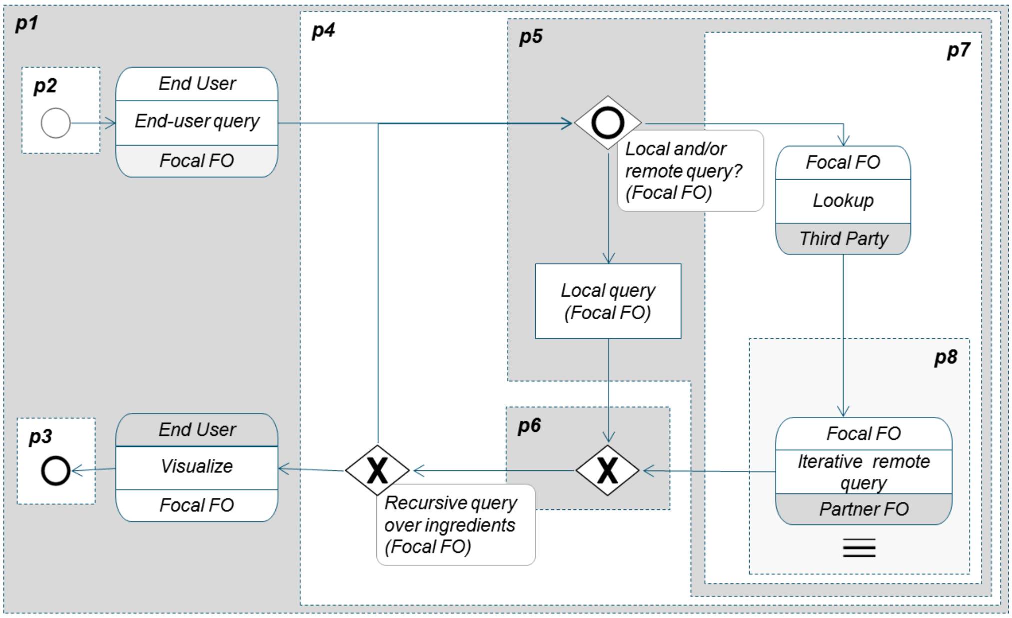


Figure 3-13: A CFP diagram for the query business collaboration process model

Table 3-7: Reference CFP allocations

CFP			*parent*								*children*						
p_{id}	*id*	*name*	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*a1*	*a2*	*a3*	*a4*	*a5*	*a6*	*a7*
p1	*1*	*sequence*									√						√
p2	*23*	*transient trigger*	√														
p3	*11*	*implicit termination*	√														
p4	*22*	*recursion*	√													√	
p5	*6*	*multi-choice*				√						√	√				
p6	*8*	*multi-merge*				√											
p7	*1*	*sequence*					√							√			
p8	*21*	*structured loop*							√						√		

3.5.2.1.2 BP2BP Alignments

The BP2BP alignments given in Table 3-8 and Table 3-9 are derived from the reference BP2BP allocation matrices and informal descriptions of the corresponding business collaboration processes.

The activity alignment matric shown in Table 3-8 is derived from the reference activity allocation matrix given in Table 3-6 in two steps. First, to represent allocations that are only part of the concrete models the alignment matrix is extended with new rows and columns. The last row (*a8*: *data capture*) is added because *a8* is only in the concrete architecture. (*a8* is also part the reference architecture but not as collaborative activity and is, therefore, not part of the reference business collaboration processes.) The *food operator* column in Table 3-6 is split into three separate columns in Table 3-8 because some activities (in this case *a8*) is misaligned only for one of the three types of food operators. Second, now that we have all the relevant rows and columns (and thus all possible allocations) of the alignment matrix we can precede to assigning alignment attributes.

We demonstrate how the alignment attributes are assigned by using the activities *a1 (end-use query)*, *a2 (decide where to query)* and *a8 (data capture)* which are associated with the *convergence*, *absence* and *divergence* alignment attributes, respectively. The allocation of activity *a1* to a third party is in *convergence* (+). The activity was allocated to a third party in the reference model. For the given case study this allocation is indeed supported in the form of the *fTrace* system that is provided by external third party. (Note, there is only one third party and thus only one concrete allocation.) The allocation of the decision activity *a2* to food operators is in *absence* (–). The activity represents a decision that a food operator has to make whether or not to look for the required transparency data from partner food operators. This activity is allocated to food operators in the reference model, but is not supported in the concrete models. The activity *a8* (data capture across food operators) is a new activity that is not present in the reference model. Therefore, *a8* represents divergent (~) behaviour. However, since not all meat processors support *a8*, the allocation of *a8* to meat processors is only *partially divergent* (#). The rest of the cells of Table 3-8 are filled in a similar fashion.

Table 3-8: Activity alignments (There is only a single third party)*

Activities	*Organisations*			
	Food operators			
	Farmers	*Meat processors*	*Bulk customers*	*Third party (*)*
query process				
a1: end-user query	x	x	x	+
a2: decide where to query	–	–	–	x
a3: local query	+	+	+	~
a4: lookup	x	x	x	–
a5: iterative remote query	–	–	–	x
a6: recursive query (ingredients)	–	–	–	~
a7: visualize	x	x	x	+
data capture process				
a8: data capture	x	#	x	~

Table 3-9 shows the how reference CFPs are aligned with their concrete counterparts. No new patterns were identified in concrete query business process models; therefore, no divergent CFPs are included. Pattern *p1*, *p2* and *p3* converge (+), the pattern *p4* largely converges except that it includes additional behaviour (*a3*) in the concrete allocation. The rest of the CFPs are missing in the concrete models since the corresponding activities are absent.

Table 3-9: CFP alignments

CFP		*parent alignment*								*children alignment*						
p_{id}	*id*	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*a1*	*a2*	*a3*	*a4*	*a5*	*a6*	*a7*
p1	*1*									+						+
p2	*23*	+														
p3	*11*	+														
p4	*22*	+										~			+	
p5	*6*				–						–	–				
p6	*8*				–											
p7	*1*					–							–			
p8	*21*							–						–		

3.5.2.2 BP2IT View

In this section we present the BP2IT allocation matrices derived from the reference business process model depicted in Figure 3-2 and the SOA service model of the generic reference architecture, which is described in the EPCIS specification and discussed briefly in section

3.5.1. We then present the BP2IT alignment matrices derived from the BP2IT allocation matrices and the informal descriptions of the relationships between concrete business collaboration processes and supporting IT systems.

3.5.2.2.1 BP2IT Allocations

Table 3-10 presents the reference IT service allocation matrix. It shows the relationships among *IT services*, *clients* (*activities*) and *providers* (*PMs*), *i.e.* BCM/DIT. These relationships are derived from Figure 3-2 and the IT services discussed in EPCIS specification.

Table 3-10: Reference IT service allocations. (FO: food operator, 3P: third party)

IT service	*PMs*				*SOA role*						
					Provider		*Broker*		*Clients*		
	App	*ETS*	*ITS*	*Disc.*	*3P*	*FO*	*3p*	*FO*	*3p*	*FO*	*End user*
QuerySrv$_{app}$	√				√						√
QuerySrv$_{ETS}$		√				√			√		
QuerySrv$_{ITS}$			√			√				√	
QuerySrv$_{discovery}$				√	√			√		√	

Table 3-11 shows the reference I/O allocation matrix. The allocation matrix shows how input and outputs data objects are allocated to activities. Note that only three data objects are considered here out of a large number of data objects, particularly, those that describe the context of an event. Most activities take an ID data object and return a list of event data objects. Some activities have special purpose, and therefore, differ from other activities in their input and output requirements. The *lookup* (*a4*) activity takes an ID and returns a list of *srvURLs*. The *recursive query* (*a6*) activity takes an event data object and returns the list of the IDs of the ingredients—if there are any. Activities involved in remote queries (*a1* and *a5*) require *srvURL* as input, in addition to ID. Visualization (*a7*) requires inputs (which are events) and master data (not included), and produces information to end-users, which is not modelled as a data object.

Table 3-11: Reference I/O allocations

Activities	*Inputs*			*Outputs*		
	ID	*event*	*srvURL*	*ID*	*event*	*srvURL*
a1: end-user query	√		√		√	
a2: decide where to query	√				√	
a3: local query	√				√	
a4: lookup	√					√
a5: iterative remote query	√		√		√	
a6: recursive query (ingredients)		√		√		
a7: visualize		√				

3.5.2.2.2 BP2IT Alignments

The BP2IT alignments given in Table 3-12 and Table 3-13 are derived from the reference BP2IT allocation matrices and informal descriptions of the relationships among IT services, clients, providers (at times also brokers), and data objects.

The IT service alignment matrix shown in Table 3-12 is derived from the reference IT service allocation matrix given in Table 3-10 by adding the required rows and columns. The new rows represent the IT services identified in the concrete case descriptions. The new columns represent the new process models and organisation types identified in the concrete case descriptions. We describe how alignment attributes are assigned using example *convergent*, *divergent* and *absent* IT service alignments.

In the reference BCM/DIT model $QuerySrv_{app}$ IT service was provided by a third party (*3P*) and used by end users (*clients*). For the given case study these allocations were indeed supported in the form of the *fTrace* app *provided* by a *3P* and used by *clients*. Both allocations are *convergent* (+). In the reference BCM/DIT model $QuerySrv_{ETS}$ IT service was provided by food operators (*FO*) and used by other *FOs* (*clients*). For the given case study these allocations were largely absent and in some cases divergent. The allocation of $QuerySrv_{ETS}$ to PM_{ETS} is *absent* (–) because $QuerySrv_{ETS}$ implements a different process model than PM_{ETS}. The allocation of $QuerySrv_{ETS}$ to provider *FO* is *absent* (–) because the FOs are not providing $QuerySrv_{ETS}$; instead, a *3P* does *divergent* (~). The allocation of $QuerySrv_{ETS}$ to client *3P* is *divergent* (~) because *client 3P* is not using *provider FOs* but *provider 3P* (potentially a different 3P than the client self). Note also that all capture IT services are new and thus all *divergent* (~).

Table 3-12: IT service alignments

IT service	*Process Models*					*SOA role*									
						Provider					*Broker*		*Clients*		
	App	*ETS*	*ITS*	*Disc.*	*BC*	*3P*	*FO*	*RG*	*V/L*	*AU*	*3p*	*FO*	*3p*	*FO*	*End user*
$QuerySrv_{app}$	+					+									+
$QuerySrv_{ETS}$		–				~	–						~		
$QuerySrv_{ITS}$			+				±							–	
$QuerySrv_{discovery}$				–		–						–		–	
$CaptureSrv_{FO}$					~		~								
$CaptureSrv_{fTrace}$					~	~									
$CaptureSrv_{QS}$					~			~							
$CaptureSrv_{HIT}$					~			~							
$CaptureSrv_{VET}$									~						
$CaptureSrv_{LAB}$									~						
$CaptureSrv_{Mynetfair}$					~					~					

The I/O alignment matrix shown in Table 3-13 is derived from the reference I/O alignment allocation matrix given in Table 3-11 by adding the required rows and columns. We describe how alignment attributes are assigned using example *convergent*, *divergent* and *absent* I/O alignment alignments.

In the reference BCM/DIT model *a1* activity takes *ID* and *srvUrl* data object and yields events. This is also how the *fTrace* app works for the given case study. Therefore, all the three allocations with reference to *a1* are *convergent* (+). The allocations in relation to the activity *a2* are *absent* (–) because the activity itself is *absent* (–, see Table 3-8). The allocations in relation to the activity *a8* are *divergent* (~) because the activity itself is *divergent* (~, see Table 3-8).

Table 3-13: I/O alignments

Activities	*Inputs*			*Outputs*		
	ID	*event*	*srvURL*	*ID*	*event*	*srvURL*
a1: end-user query	+		+		+	
a2: decide where to query	–				–	
a3: local query	+				+	
a4: lookup	–					–
a5: iterative remote query	–		–		–	
a6: recursive query (ingredients)		+		+		
a7: visualize		+				
a8: data capture					~	

3.5.2.3 IT2IT View

In this section we present the IT2IT allocation matrices derived from the reference model for the distributed IT as discussed in the EPCIS specification (see also section 3.5.1). We then present the IT2IT alignment matrices derived from the IT2IT allocation matrices and the informal descriptions about the distributed IT.

3.5.2.3.1 IT2IT Allocations

IT2IT viewpoint specifies *IT system*, *data object* and *DFP allocations*. Since neither the reference architecture nor the descriptions of the concrete architectures provide information about IT system models, IT system allocations and the corresponding alignments are not included.

Table 3-14 presents the reference data object allocation matrix. It shows how the reference data objects are allocated to organisations. The allocations are derived from the ECPIS specification, which is briefly discussed in section 3.5.1. Though a great number of data objects, particularly involving master data, may be involved, we considered only products identifications (*IDs*), transparency data items (*events*) and service addresses (*srvURLs*), which are the three key data objects of the reference architecture. Their allocation is simple: food operators manage their own event data objects; the third party manages the service addresses. Both actors manage IDs for different purposes in the query PMs: food operators resolve ID to events, while the third party resolves ID to srvURLs.

Table 3-14: Reference data object allocations

Data objects	*Organisations*	
	Food operators	*Third party*
ID	√	√
event	√	
srvUrl		√

Table 3-15 shows the reference DFP allocation matrix. DFP allocation matrix has three dimensions: *data object*, *DFP* and *organisation*. According to the reference architecture the allocation of DFPs to data objects is not dependent on the organisation; therefore, the reference allocation matrix does not include the organisation dimension. The allocation matrix can consist of a maximum of 40 rows, one for each DFP. A data object is associated with at least four DFPs; one from each category of DFPs. The reference DFP allocations are discussed based on the four categories.

Visibility: IDs are allocated *activity scope* (*dfp1*) DFP because an ID is obtained from end-user and is passed from one activity to the next. Events and srvURLs are, in comparison, allocated *external data scope* (*dfp-8*) DFP because they are fetched from repositories that are external to the process orchestration system.

Interaction: IDs are allocated *activity-to-activity* (*dfp*-9) DFP because IDs are passed from activity to activity. Events and srvURLs are allocated *activity pulls data* (*dfp-16*) DFP because activities pull data from external EPCIS repositories.

Transfer: All data objects are allocated *pass inputs by value* (*dfp-27*) and *pass outputs by value* (*dfp-28*) transfer DFPs.

Routing: IDs do not affect the routing of the control flow; therefore, no routing DFPs are assigned to them. *Events* are assigned *data-based routing* (*dfp-40*) DFP because the content of an event data object determines if recursive queries are executed. S*rvURLs* are allocated *value as activity post-condition* (*dfp-36*) routing DFP since without a service address external queries cannot be executed.

Table 3-15: DFP allocations

Categories of DFPs	*DFPs*	*Data objects*		
		ID	*event*	*srvURL*
Visibility	*1*	√		
	8		√	√
Interaction	*9*	√		
	16		√	√
Transfer	*27*	√	√	√
	28	√	√	√
Routing	*36*			√
	40		√	

3.5.2.3.2 IT2IT Alignments

The data objects alignment matrix is simple. The allocations of ID and event data objects to organisations are *convergent* (+) since for the given case study these data objects are allocated as defined in the reference data allocation matrix. The allocation of srvURL to third party organisation is absent (–) since for the given case study there is only a single srvURL value. The data object alignment matrix is trivial and, therefore, it is not presented.

The DFP alignment matrix shown in Table 3-16 is derived from the reference DFP allocation matrix given in Table 3-15 by adding the required rows. The new rows are the DFPs that are identified in the concrete case descriptions. The reference allocation of DFPs to data objects is not dependent on the organisation but the concrete allocations can differ from organisation to organisation. Such variations are however not considered in the table. We describe how alignment attributes are assigned using example *convergent*, *divergent* and *absent* DFP alignments.

In the reference data sharing model IDs are allocated *activity scope* (*dfp1*) DFP. For the given case study also IDs have activity scope. Therefore, the given DFP allocation is *convergent* (+). In the reference data sharing model events are allocated *activity pulls data* (*dfp-16*) interaction DFP. For the given case study the events are not pulled from external transparency data repository. Note that, in the given case study, the focal food operator captures all transparency data and serves query request from own local repository. Therefore, the given DFP allocation is *absent* (–). In the reference data sharing model data capture is a local process. For the given case study data capture is a collaboration process. All DFP allocations associated with new collaboration processes are considered *divergent* (~).

Table 3-16: DFP alignments

Data flow concerns	*DFPs*	*Data objects*		
		ID	*event*	*srvURL*
Data query business collaboration				
Visibility	*1*	+		
	8		+	–
Interaction	*9*	–		
	16		–	–
Transfer	*27*	–	–	–
	28	–	–	–
Routing	*36*			–
	40		–	
Data capture business collaboration				
Visibility	*1*			
	8		~	
Interaction	*9*			
	16		~	
Transfer	*27*		~	
	28		~	
Routing	*36*			
	40			

3.5.3 Using the Alignment Matrices

In a previous research (which is chapter 4 in this thesis) we have developed a reference architecture for Chain-wide Meat Transparency System (CMTS) from the generic EPCIS reference architecture (Kassahun *et al.* 2014). We also realized a transparency system based on CMTS and demonstrated the system (Kassahun *et al.* 2016, which is chapter 5 in this thesis). The main elements of CMTS are reproduced in Figure 3-14. In the following we discuss how the alignment matrices of the previous section explain the development of CMTS.

The EPCIS standard (GS1 EPCglobal 2014) specifies a distributed network of internal transparency systems (ITS). The standard provides fixed IT interfaces ($\text{QuerySrv}_{\text{ETS}}$ and $\text{QuerySrv}_{\text{ITS}}$) and data models (the EPC-related event data models) that the ITSs have to comply with in order to be part of chain-wide transparency systems. In terms of the EPCIS standard all food operators are similar in their ability to realize EPCIS-compliant ITSs. However, many food operators cannot comply with the specification of EPCIS in various ways. The alignment matrices given in tables 3.8, 3.9, 3.12, 3.13 and 3.16 show how the food operators *converge* (*i.e.* how they comply), fail to support (*absence*, *i.e.* how they fail to comply), and *diverge* (*i.e.* how they support their requirements in their own ways).

We provide below few examples of how explicit statements about alignment facilitate the process of adapting the EPCIS reference architecture into CMTS. Many food operators cannot, for instance, support some of the required activities as given in the activity alignment (Table 3-8) matrix. As a result many of them cannot provide the required IT services

(QuerySrv$_{ETS}$ and QuerySrv$_{ITS}$, see IT service alignment matrix given in Table 3-12). To address these problems one of the food operators of the supply chain (of the given case study) have introduced new business collaboration activity (*a8* in Table 3-8), which help the third party to support QuerySrv$_{ETS}$ using a different strategy to realizing chain-wide transparency. Other food operators were willing to use other ways of realizing ITSs, for instance, using an on demand cloud-based ITS system.

The alignment matrices help analyse the requirements of the food operators and their capabilities, which in turn are governed by many other factors, such as the requirements of the food regulations and the availability of new IT platforms. Based on the alignment we differentiated five types of food operators in CMTS. We added new activities and incorporated new IT system allocation strategies by considering the roles third parties, who already play a major role in realizing chain-wide transaparency in some meat supply chains. By differentiating five types of food operators in CMTS we were able to suggest five different solutions for a given problem. For insatnce, the activity *a2* (in which a food operator determines where to query transparency data from, *i.e.* locally, remotely or both) is *absent* (–). As a consequence, the activities *a5* (iterative remote query), which constitute the major part of *external transparency* is also *absent*. It turns out that these misalignments cannot be addressed in a uniform manner. For some food operators (such as farmers) the solution is using a transparency systems as a service such as that demonstrated by Kassahun *et al.* (2016, chapter 5). For others, the solution is an integration APIs for legacy systems provided by a third party. The various solutions are discussed in detail in chapters 4 and 5.

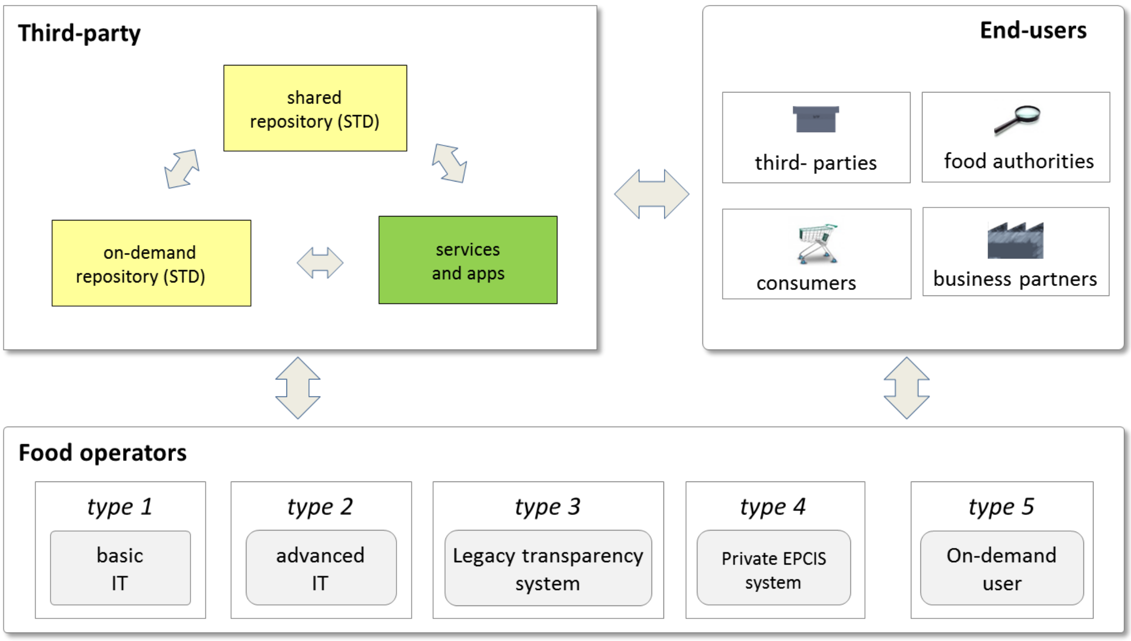


Figure 3-14. The elements of a chain-wide meat transparency system (adapted from Kassahun et al. 2014)

3.6 Discussion

The problem of business-IT alignment has been broadly addressed in literature in the context of a single organisation and less attention is dedicated to the alignment concerns of multiple collaborating organisations. In this paper we have focused on the alignment concerns of multiple collaborating organisations. To address the alignment concerns more explicitly we have offered a business-IT alignment framework called BITA* and demonstrated the framework in a real-life business case. In this respect this paper complements the existing literature while also addressing novel concerns that were not explicitly addressed before.

In BITA* we distinguish three types of alignment concerns: BP2BP, IT2IT and BP2IT. BP2BP refers to the alignment of business collaboration process models. IT2IT refers to the alignment of the models for the distributed IT. BP2IT refers to the alignment of the models that specify how the business collaboration processes should be supported by the distributed IT.

Alignment refers to the fact that each collaboration organisation has its own models for addressing alignment concerns and these models need aligning. To support the alignment process we provided required alignment modelling abstractions and a systematic approach for applying alignment models. An important aspect of the alignment modelling is the adoption of workflow patterns that appeared to be very valuable in enabling model comparison. The alignment models are organised in three coherent design viewpoints. The three viewpoints are BP2BP, BP2IT and IT2IT viewpoints and have been carefully designed according to formal viewpoint design guidelines (ISO/IEC/IEEE 2011).

The approach has been applied in a real-life business case on a transparency system for meat supply chains. The case included a large and diverse number of organisations including farmers, meat processors, bulk customers of meat products (such as retailers and caterers) and third parties. It appeared that each of these stakeholders differed with respect to the adopted architectures, collaboration concerns, business processes, and IT systems. Yet, our approach was successfully applied to model the alignments of the diverse business collaboration models involved in realizing supply chain wide transparency systems.

Alignment modelling has been used in BITA* to help derive reference architecture for the given problem (such as chain-wide transparency for meat supply chain) from a generic and broadly-accepted reference architecture. For the case study the adopted generic reference architecture was the EPCIS which appeared helpful. There are, however, cases in which a suitable generic reference architecture is unavailable. For such cases, further research is required on how to design a bootstrap reference architecture that will help the alignment process.

While applying the approach, we could observe the following. First of all, adopting an explicit process for the alignment problem is very helpful to support awareness of the alignment problems and likewise to create a common understanding among the stakeholders. Because of the explicit alignment process the alignment problems could be more easily identified and the relevant reference architecture adapted before the collaborating partners start the often difficult process of redesigning business process and IT systems. This was vital because misalignment identified later in the process of system development would be more problematic and costly for all stakeholders. Second, the viewpoints that we have provided seemed to be necessary to make the alignment process explicit and model-based. So far, the adopted models were basically using existing business process and IT design abstractions. No explicit design abstractions were provided for representing alignment concerns. As such, the overall communication of these concerns and the guidance for solving them were seriously limited.

We have focussed in this paper on alignment concerns of multiple collaborating organisations. The BITA* approach is, however, equally applicable when only two or few organisations are involved, in which case the collaboration architecture adopted by one of the organisations serves as a reference architecture. This is usually the case when dependent organisations (for instance, suppliers) must align their business processes and IT systems with the dominant (focal) organisation (such as a large manufacturer or a large retailer).

3.7 Related Work

There is a considerable literature on alignment. However, explicit alignment models and an accompanying design framework are lacking. In this section, we provide only a short overview of related work that focus both on modelling abstractions and systematic approaches.

Chen et al. (2005) proposed BITAM (Business IT Alignment Method): BITAM, which is a systematic approach consisting of twelve steps for detecting and correcting misalignments. Typical steps of their approach are for instance, *elicit business and IT architecture from architects* (step 5 and 6), *map operational scenarios onto business and IT architectures* (steps 7 and 8), and *assess the misalignments* (step 9). However, the approach depends on personal perceptions and not on explicit models. Recently, Hinkelmann et al. (2016) propose a business and IT alignment approach that combines enterprise architecture modelling (including the modelling approaches we used in this paper) and enterprise ontologies. Enterprise ontologies are tools of knowledge engineering and enable explicit specifications of conceptualizations of a given problem domain (Gruber 1995) and will enable building a knowledge base of explicit representation of reference models and patterns. However, the authors did not present the required knowledge base that is comparable to the workflow patterns that we successfully applied. Yet another recent related work focused on a specific aspect of misalignment between business processes and software user interfaces (Hoch *et al.* 2016). Hoch *et al.* accurately identify the fact that gaps between business processes and their supporting software exist because the representation of process elements in the software models is implicit – a case in point being the lack of user interface specification in BPMN models. They have, therefore, proposed a model of representing business artefacts to enrich BPMN models so that implicit assumptions of business process and the unforeseen business-IT misalignments can be avoided. In addition to these and other business-IT alignment approaches the existing enterprise designs frameworks, such as TOGAF (The Open Group 2011), provide methodologies that address alignment issues. However, these frameworks can be characterised as a one way alignment methods because they guide how to design the IT to fulfil the requirements laid out in the form of business process models, but not the other way round. Generally, the common limitation of existing alignment approaches and enterprise design frameworks is that they do not address alignment problem in which multiple organisations are involved.

3.8 Conclusion

In this paper we have presented BITA*, a framework for aligning business processes and IT systems of multiple collaborating organisations. We identified three types of alignment concerns including business process to business process (BP2BP), IT to IT (IT2IT) and business process to IT (BP2IT) alignment concerns, and provided the corresponding three alignment design viewpoints.

Recognizing the difficulty of comparing incomparable models of business process and IT from diverse organisations, we introduced a number of key concepts in BITA* that has allowed us to develop models for aligning business process and IT models in a uniform manner. First, we used reference models as a common model with which diverse models from diverse organisations can be compared. Then we introduced allocation models as means of uniformly representing diverse business process and IT models that have to be aligned. Third, we used workflow patterns to support capturing complex business process and IT models as allocation models. These conceptualizations enabled us to design alignment models. The alignment models include explicit alignment attributes—convergence, absence, divergence, partial convergence, partial divergence, irrelevant—that can be assigned to allocations.

We presented a step-by-step approach that shows how business analysts and software architects can align the diverse concrete models with the reference models iteratively, and how they can incrementally improve the concrete and reference models until the desired level of alignment is achieved. Finally, we demonstrated the framework by applying it to an industrial case study.

In our future work we aim to build a design support system for further assisting the business analysts and architects in the modelling and alignment process. A relevant future study in this context could be the enhancements of workflow patterns for recurring business collaboration concerns as the workflow patterns that we used were originally devised to describe centralized workflow systems of individual organisations.

ACKNOWLEDGEMENTS

The research leading to this paper received funding from the European Community's Horizon 2020 Programme (H2020/2014-2020) within the IoF2020 project under grant agreement No. 731 884. We acknowledge the European Community for the funding. We would also like to acknowledge Rob Hartog, former member of the Information Technology Group of Wageningen University, for the careful review of the paper.

4

DERIVING A REFERENCE ARCHITECTURE FOR TRANSPARENCY SYSTEMS IN MEAT SUPPLY CHAINS [1]

[1] Published in the journal of Computers and Electronics in Agriculture (ISSN: 0168-1699): Kassahun, A., R. J. M. Hartog, T. Sadowski, H. Scholten, T. Bartram, S. Wolfert and A. J. M. Beulens (2014). Enabling chain-wide transparency in meat supply chains based on the EPCIS global standard and cloud-based services. *Computers and Electronics in Agriculture*.

Abstract

Transparency in meat supply chains is necessary to guarantee the safety, quality and trust of consumers in meat products. However, transparency systems currently in place are often not adequate for sharing transparency data among food operators, providing consumers accurate transparency information, or enabling authorities to respond quickly and effectively in cases of food safety emergencies. Due to major meat crises and scandals the meat sector has in this respect attracted substantial attention. In this paper we identify regulatory, business, consumer and technological requirements for meat supply chain transparency systems and present a reference software architecture that will guide the realisation of these systems. The reference architecture is characterized by three main elements: the EPCIS standard for tracking and tracing, cloud-based realisation of transparency systems, and the provision of transparency systems as services by third-party transparency service providers (3pTSPs). Usage scenarios are presented to explain how the different types of meat supply chain actors can use transparency systems that are based on the architecture.

Keywords: Meat supply chain transparency; Consumer awareness in meat; Tracking & tracing; Food regulations; EPCIS; Cloud-based services

4.1 INTRODUCTION

Transparency in meat supply chains is necessary to guarantee the safety, quality and trust of consumers in meat products. Consumers' trust in meat products, production, origin and the actors[2] involved is crucial for the functioning and competitiveness of local, regional and global food markets (Brom 2000, Schiefer 2011). Particularly meat is a relatively sensitive product as highlighted by major crises and scandals such as the BSE (Bovine Spongiform Encephalopathy, commonly called mad cow disease) crisis (Collee and Bradley 1997), the dioxin crisis (Verbeke 2001) and the recent horse meat scandal (Premanandh 2013). As a result a number of transparency measures are incorporated in food regulations such as the European regulation Reg. N^{o} 178/2002 (also referred to as the General Food Law - GFL) and the more recent regulation Reg. N^{o} 1169/2011.

Crucial aspects of transparency are tracking and tracing (traceability) and the ability to make consumers[3] aware of a wide range of quality attributes of their food. Traceability refers to the ability to *track* downstream the supply chain where a distinct batch or lot of product is (or is being processed) and to *trace* upstream the supply chain from where a distinct batch or lot came (van Dorp 2004). In this article 'consumer awareness' refers to awareness of consumers about the diverse quality attributes of the meat products they buy, such as, nutritional value, place of origin or provenance, ingredients, specific quality attributes, and allergy risks.

Today's transparency systems rely largely on basic technologies, mainly, labelling and "paper trails" left by email, fax or EDI (Electronic Data Interchange) business interactions. Some large meat processing companies do have transparency systems in place as part of their

---

[2] We use the terms *actor*, *company*, *food operator* and *business* interchangeably in this article.

[3] We use the term *consumer* to mean shoppers as well as consumers at home or elsewhere.

enterprise system, however, the use of state-of-the-art enterprise transparency systems rarely extends entire meat supply chains (Trienekens *et al.* 2012).

Consumers rely almost exclusively on labels for information about meat products they buy from retailers. A label is a printed tag that is physically attached to the product and the information it carries can only be accessed if one can physically get hold of the product. The dependence on labels can be ascribed to the requirements of food regulations that mandate them as exclusive means of communication with consumers. Food regulations do not yet cover remote access to transparency information even though the Internet is a commonplace in today's society and consumers increasingly rely on it for information. A notable exception in this respect is the recent European food regulation, Reg. N^{o} 1169/2011, that goes beyond labelling and towards rules that govern other means of access to food information, including the Internet (article 27, EC 2011).

Food regulations have also major influence on the way food operators collaborate and exchange information. In Europe, GFL prescribes the one-back/one-forward principle to meat transparency. According to this principle food operators are only required to identify and share information with their immediate suppliers (one-back) and immediate customers (one-forward) (EC 2007, EC 2011). This leads to a linear one-back/one-forward collaboration chain where transparency data is passed to retailers through successive links from farms and across the various intermediate actors (*i.e.* slaughterhouses and meat processing companies). This method is however not robust because, in practice, not all food operators implement state-of-the-art transparency systems and the benefits of gathering detailed transparency data by one actor are largely lost when subsequent actors are not able to pass on the data.

Realizing chain wide transparency – for either addressing food safety emergencies or enhancing consumer awareness – requires that each individual food operator implements a transparency system inside its production facilities, and that information flows smoothly among the individual transparency systems. Chain-wide transparency systems can thus be considered to consist of two complementary sub systems – internal and external transparency systems (Moe 1998, Gandino *et al.* 2009). Realizing internal transparency requires food operators to establish the logical links between the identification code of a specific batch of output products they deliver to their customers to the identification codes of specific batches of input products (ingredients) they obtained from their suppliers and used in the making of the output products. Realizing external transparency requires pairs of trading food operators to establish the logical links between identification codes of products delivered by the one and received by the other.

For food operators to engage in an efficient and effective information exchange their internal transparency systems should be based on electronic record keeping and the information exchanged should conform to standards. The need to share traceability data across a wide range of industries led the GS1, a global consortium of businesses, to develop the EPCIS (Electronic Product Code Information Services) standard (GS1 EPCglobal 2014). The standard specifies how traceability data are captured digitally and defines standard data types and interfaces for exchanging them. The information exchanged is about individual or a class of product items that are uniquely identified globally by an identification code called EPC (Electronic Product Code).

Chain-wide transparency systems can be realized using a linear, centralized or distributed model of collaboration (Meuwissen *et al.* 2003, Folinas *et al.* 2006, GS1 2010, Bhatt *et al.* 2012). An example of the *linear model* of collaboration is the one-back/one-forward approach. In the *centralized approach*, such as national bovine animal registration systems in Europe (EC 2000, EC 2004), a shared transparency system is created where transparency data is collected and from which it is accessed. In the *distributed approach* food operators maintain own transparency systems that are interconnected into a network. One approach to realize a distributed model of collaboration is to adopt the EPCIS standard (Shanahan *et al.* 2009, Thakur *et al.* 2011).

Recent experiences in practice as well as research literature indicate that, besides the one-back/one-forward method, both the centralized and the distributed scenarios are viable forms of collaboration for realizing chain-wide transparency systems (Bowling *et al.* 2008, Myhre *et al.* 2009, Shanahan *et al.* 2009, Hartley 2013). However, besides the national (centralized) bovine animal registration systems and few experimental distributed (EPCIS-based) systems we are unable to determine a widespread use of these two approaches. The centralized approach is simple to implement but requires either trust among supply chain actors or regulatory mandate. In addition the centralized approach requires a trusted third-party that manages the centralized system to which all food operators will have to publish transparency data. Distributed systems, on the other hand, require that each food operator maintains state-of-the-art transparency system following global standards (such as the EPCIS standard). But, state-of-the-art systems are costly and in most cases beyond the means of small businesses.

In recent years, the cloud computing paradigm is enabling standard software packages to be available as a service following the SaaS (Software as a Service) business model. This new business model makes state-of-the-art software affordable and accessible on-demand over the Internet. The European Future Internet Public-Private Partnership (FI-PPP) programme (FI-PPP 2013) aims to accelerate the adoption of this new Internet-centric technologies in Europe by providing the building blocks required to realize the technologies.

In this paper we argue for such a cloud- and standards-based approach for realizing chain-wide transparency systems. We further argue that these systems have to accommodate both centralized and distributed forms of information sharing and collaboration. We present a reference architecture that shows how this can be achieved.

The paper is organised as follows. In section 2, we describe the methodology followed. In Section 3, we discuss the current state of transparency systems in meat supply chains with the help of an illustrative example. In section 4, we formulate a number of requirements for the reference architecture. In section 5, we present the reference architecture based on the requirements outlined in section 4. Finally we make concluding remarks in section 6.

4.2 RESEARCH APPROACH

The work presented in this paper is design-oriented research conducted in the context of two research projects: Smart Agri-Food (SAF) (SAF 2013) and its follow-up FIspace (FIspace 2013). Both projects are part of the European FI-PPP program. In this program a new integrated IT infrastructure is being developed and tested in three phases. At the core of FI-PPP is the FI-Ware project (FI-Ware 2013) that develops a core platform consisting of a set of IT Generic Enablers (GEs). Around the FI-Ware project are a number of use case projects, in which requirements are gathered and the resulting platform tested (Brewster *et al.* 2012). SAF and FIspace are two of such use case projects. This paper is based on a pilot study within SAF and FIspace in which the architecture is designed.

The design process is done in three steps. First, we analysed the current state of meat supply chain transparency. The analysis is based on a beef supply chain in Germany (hereafter referred simply as *the supply chain*), which we consider to be representative of major meat supply chains in Europe. To gain insight two focus group workshops were conducted in November and December of 2011 involving representatives of relevant organisations, food operators, retailers, and members of the FIspace research team. We also visited a large slaughterhouse (hereafter simply referred to as *the slaughterhouse*) that is part of the beef supply chain of our pilot study. The organisations involved in the workshops include GS1, Orgainvent, EHI, Global G.A.P., QS, the slaughterhouse and two supermarket chains in Germany. GS1 is a global not-for-profit organisation that is responsible for developing global standards to improve the efficiency and transparency of supply chains; Orgainvent is an organisation responsible for standardizing meat labelling in Germany; EHI is a scientific institute of the German retail industry; Global G.A.P is a global organisation that promotes good agricultural practices and QS is an independent meat quality assurance company in Germany.

Second, we identified a number of requirements through the workshops and subsequent formal and informal contacts with the representatives. The workshops were followed by facility visits of a meat processing plant and informal interviews of the representatives of the slaughterhouse, the meat processing plant and two of the major supermarket chains in 2012. The visits, interviews and the materials we received provided us with detailed information about the processes in meat supply chains. Besides, we received additional information in bilateral correspondence with the representative of the slaughterhouse, Global G.A.P, GS1 and other relevant organisations to obtain a rich appreciation (Checkland and Winter 2005) of the state of transparency in meat supply chains and formulate possible improvement options.

Last, we designed the reference architecture incrementally and iteratively following the requirements identified in the previous step. We employed usage scenarios to demonstrate how a transparency system based on the reference architecture can be utilized and improved on the design using the insights gained. Even though the architecture builds mainly on the FIspace platform our aim is to provide a reference architecture that will serve as a blueprint for future meat transparency systems on any comparable cloud-based platform.

4.3 The current state

Transparency systems in today's meat supply chains are too diverse to make a general description. We, therefore, use our pilot study as an illustrative example of the current state. The example is from a beef supply chain in which the slaughterhouse involved is the focal company.

4.3.1 An illustrative example

The supply chain from the perspective of a representative of the slaughterhouse is depicted in Figure 4-1. The figure shows that the slaughterhouse plays a key role in the flow of transparency information; it shows how the slaughterhouse gathers data from farmers and passes them along the flow of products to the various downstream actors. The slaughterhouse and farmers share data with third parties too (orthogonal to the flow of products). The third parties involved are the QS quality assurance agency, the HIT national bovine animal registration office, veterinaries, laboratories, provider of a trade fair web portal (Mynetfair) and an independent third-party (hereafter referred to as *the 3pTSP – the third-party transparency service provider*) that provides a transparency system called fTRACE. fTRACE is a subject of this paper and will be described in detail in the next section.

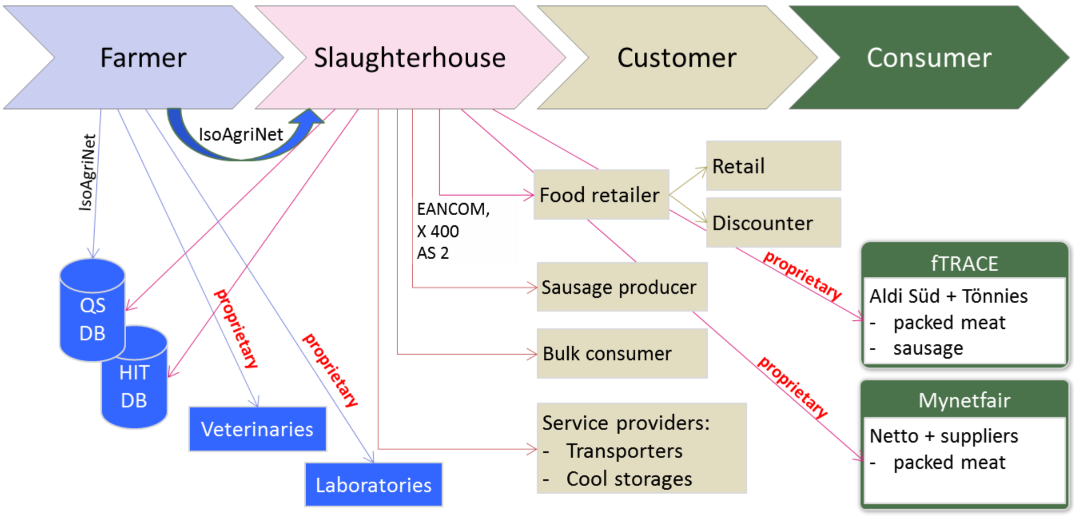
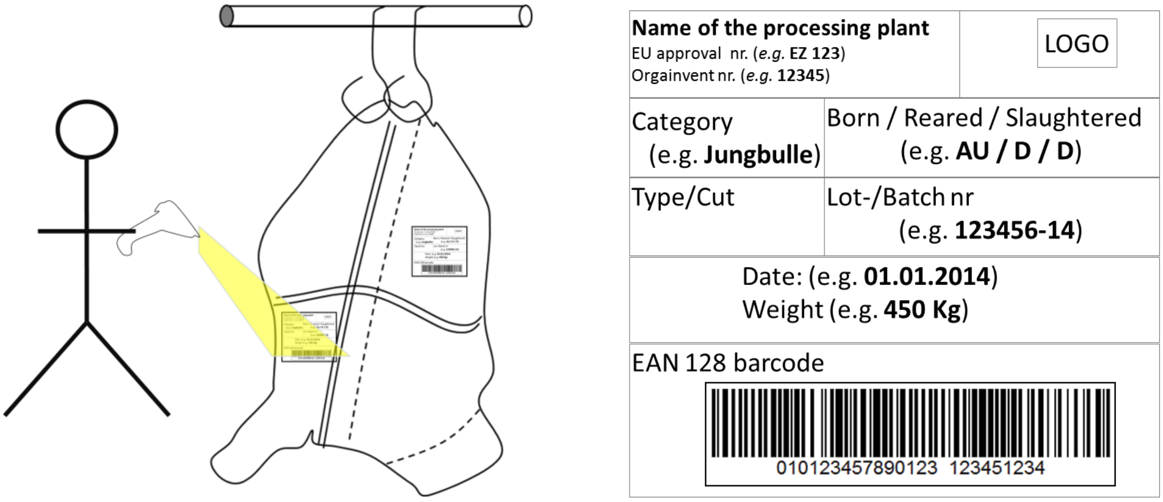


Figure 4-1. Data flow in a German meat supply chain (courtesy of the representative of the slaughterhouse; translated from German). QS (Qualität und Sicherheit) is a German quality assurance scheme with an associated company by the same name that does the bulk of meat quality assurance audits in Germany(Albersmeier et al. 2009). HIT (Herkunftssicherungs- und Informationssystem Tiere) is a German national database for registration of movement of bovine animals established in accordance with the EC Regulation 1760/2000 (EC 2011). Mynetfair is a trade fair web portal (Mynetfair 2013). fTRACE is a third-party meat transparency system offered by GS1 Germany (fTRACE 2013).

Within the slaughterhouse labels are used during many of the internal processing steps in order to comply with European regulatory prescription (see Figure 4-2). The labels are standardized using the Orgainvent voluntary labelling scheme. An internal transparency system is realized using an ERP (Enterprise Resource Planning) system.

4.3.2 fTRACE

A system that is used in *the supply chain* but that is not representative of the current state elsewhere is the fTRACE system; in fact it is unique in Europe. fTRACE is a web-based third-party transparency system consisting of a smartphone application (the fTRACE smartphone app) and a transparency database managed in the fTRACE server. The database is regularly updated by the slaughterhouse using data from its own operations and its suppliers, which are mainly farmers. The fTRACE system works as follows. A QR-code is printed on the meat package alongside other labelling information. The code encodes the web address (URL) of the fTRACE server and the unique identification code (ID) of the batch from which the meat product comes. A user scans the QR-code in supermarkets or at home with his or her smartphone. The smartphone decodes the QR code into a web address and a unique batch number, fetches transparency information from the fTRACE server and presents the information to the user in a user-friendly form. The user interface of the fTRACE smartphone app used during testing in a Spanish supermarket in connection to this study is shown in Figure 4-3.



(a) (b)

Figure 4-2. An example of an intermediate meat product i.e. a carcass quarter (a) and paper (printed) label placed on the intermediate product (b) (based on real-life images obtained from the slaughterhouse)

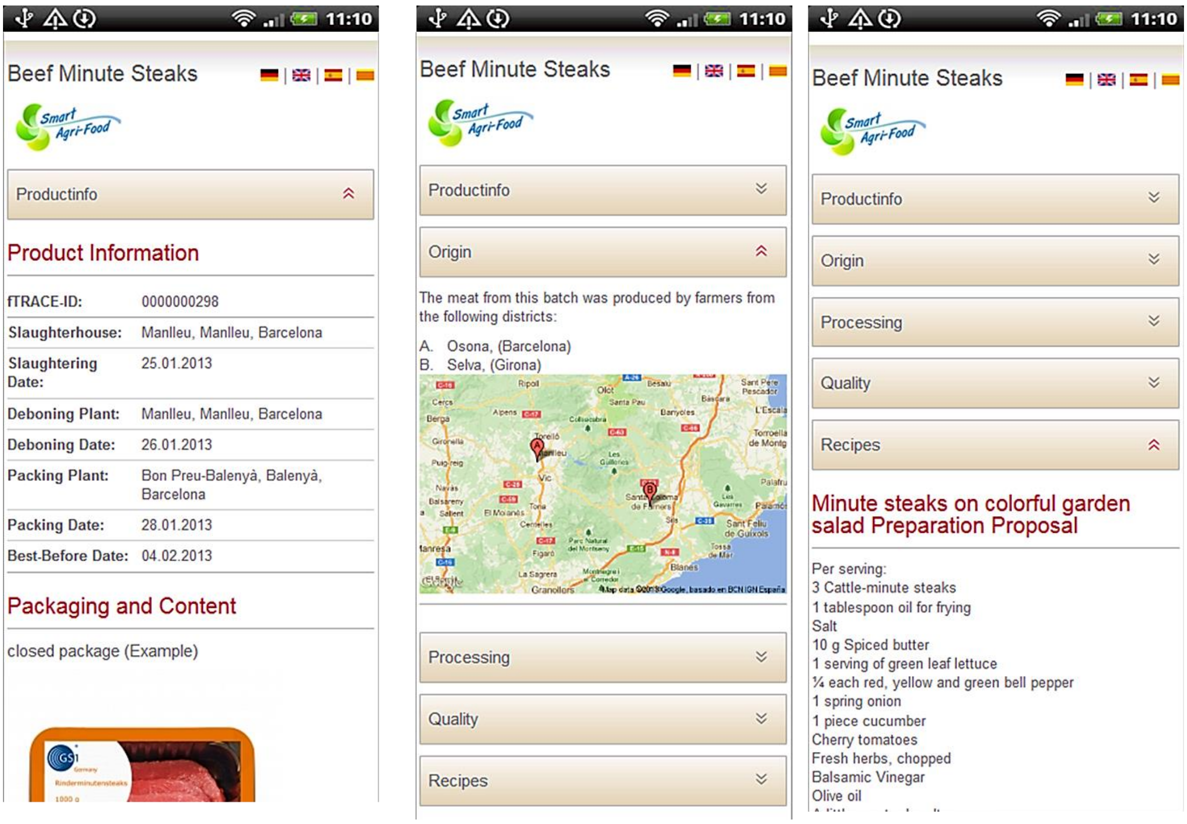


Figure 4-3. fTRACE mobile app (version of January 2013 being tested in a Spanish supermarket)

fTRACE was originally commissioned by a meat processing company for use by its own consumers. Realizing that its use is better managed by an independent and trusted third party the meat processor transferred fTRACE to the 3pTSP. Since then fTRACE has been used in a number of supermarkets in Germany.

4.3.3 The architecture of the transparency system

We describe the architecture of the transparency system of the illustrative supply chain by starting with a 'walk through' the system and continue with an analysis of its key aspects.

4.3.3.1 A walk through the current transparency system

Figure 4-4 depicts the supply chain under consideration along with the flow of information through the transparency system. The transparency system consists of two information flow channels through which transparency data travel. The first is a paper-based information channel wherein transparency data are passed via labelling and delivery notes. The second is a digital information channel wherein transparency data are transmitted electronically.

Chain-wide transparency starts with capturing transparency data **(1)** (see Figure 4-4) either by scanning a label on the product (as in Figure 4-2) or entering data manually. The data captured are then stored in the information system of the food operator. The slaughterhouse uses an advanced ERP system; farmers use diverse basic (or desktop) information systems; other food operators use various types of systems **(2)**. When a meat product is processed and passed to the next food operator, the meat is labelled and the delivery is usually accompanied by a paper delivery note **(3)**. At the same time order and delivery information is transmitted electronically **(4)**. The product is finally delivered to retailers. In the retail shop the label on the product is the main source of transparency information **(5)**. In the supply chain under consideration data is also submitted to the 3pTSP by *the slaughterhouse* **(6)**. In such cases users can access detailed transparency information with their smartphones **(7)**.

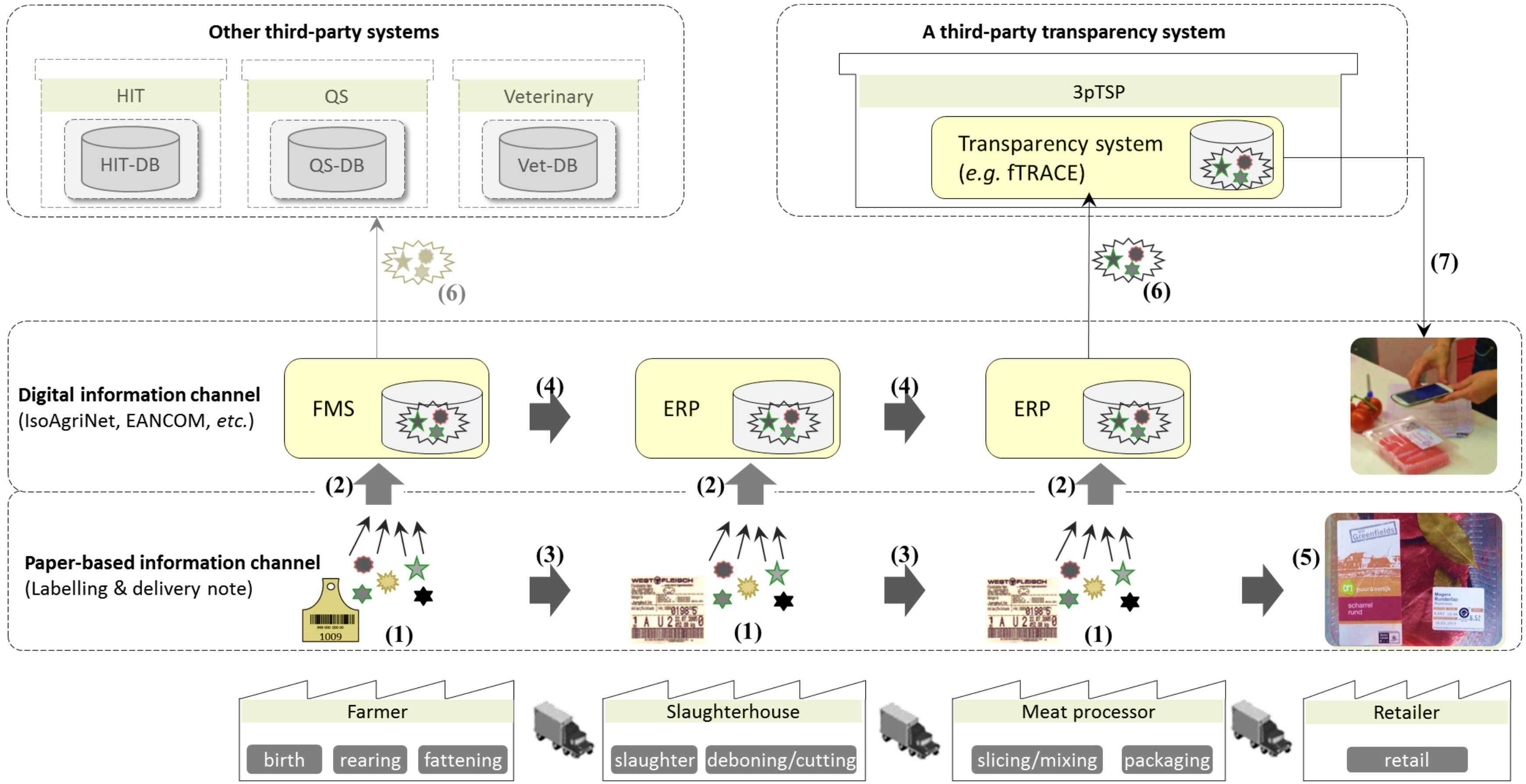


Figure 4-4. The transparency system of the meat supply chain of the pilot study. (FMS = Farm Management System, DB = Database/Repository). The figure is explained in detail in the text.

4.3.3.2 Key aspects of the system

Based on the above descriptions we identify six key aspects of the system. The first aspect relates to how information is provided to consumers (and shoppers). Clearly, the last meat supply chain actor responsible for the packaging and labelling of the meat product is the information provider. The supply chain under consideration is in this respect different because the slaughterhouse is able to provide transparency information digitally to consumers with the help of a 3pTSP using a smartphone app. As a result, the ease with which information is provided to consumers and the level of detail of the information is significantly different from label-based information provision.

The second aspect relates to the way food operators collaborate to realize chain-wide transparency. Dictated by food regulations they collaborate and exchange information only with their direct suppliers (one-back) and direct customers (one-forward). As a result food operators normally have no way of asking for information directly from the suppliers of their suppliers and they also don't have the means of reaching the customers of their customers directly. Again, the supply-chain of our case is an exception, since the slaughterhouse can reach the customers of its customers (i.e. shoppers at supermarkets) through the fTRACE system.

The third aspect relates to data sharing. Food operators share data among each other using a diversity of data transfer methods, including labels, emails, fax and EDI. Businesses communicate using such communication protocols as X.400 (ITU-T 1999) and AS 2 (Moberg and Drummond 2005) but the main purpose of communication is for handling business transaction and not exchanging transparency information. Transparency data have to be filtered from the business transaction data.

The fourth aspect relates to data format. Two food operators can share information only if they use a shared data format and semantics. In the supply chain both standardized and proprietary data formats are used. Farmers pass data to the slaughterhouse using the ISOagriNET standard (ISOagriNet 2013); the slaughterhouse uses the EANCOM standard (GS1 2013) to communicate with its customers. Both farmers and the slaughterhouse use a number of specialized or proprietary formats to communicate with third parties, including the fTRACE system.

The fifth aspect is related to data storage. Each food operator is required to keep a record of transparency information according to the requirements of GFL. Food operators do keep a record of data required by regulations but not always in a digital form – as the regulatory requirements do not demand electronic record keeping. In fTRACE, transparency information is stored digitally both in the information system of the slaughterhouse and at the 3pTSP. Storing data digitally at 3pTSP has been a sensitive issue for the slaughterhouse and it may in future not transfer bulk data to the 3pTSP but allow it to query data on-demand.

The sixth aspect refers to access to information, particularly, by regulatory authorities in case of food safety emergencies. Authorities have the right to obtain information from all actors in cases of emergencies but there is currently no standard or specification we know of that allows them to query data across the entire meat supply chain electronically and quickly in a uniform manner. The fTRACE system can provide an effective means of tracking and tracing in case of food safety emergencies but this aspect has so far never been tested or used.

4.4 ARCHITECTURAL REQUIREMENTS

We categories architectural requirements for future chain-wide meat transparency systems into regulatory, business, consumer and technological requirements. These requirements are described below.

4.4.1 Regulatory requirements

Food regulations impose specific requirements on how transparency systems shall function because they cover aspects of record keeping and data sharing – besides quality and safety aspects. These aspects pose specific IT requirements on transparency systems. We mention only the most significant EU food regulations in this regard.

EU regulation Reg. N^{o} 1760/2000 (EC 2000) requires the identification and registration of bovine animals and the labelling of beef and beef products. Paragraph 1 of article 13 of the regulation particularly states: “The compulsory labelling system shall ensure a link between, on the one hand, the identification of the carcass, quarter or pieces of meat and, on the other hand, the individual animal or, where this is sufficient to enable the accuracy of the information on the label to be checked, the group of animals concerned”.

The General Food Law (EC 2002) and its amendments mandate the one-back/one-forward traceability model. The commission clarifies the essence of this principle as: “... the requirement for traceability is limited to ensuring that businesses are at least able to identify the immediate supplier of the product in question and the immediate subsequent recipient” (EC 2007).

Though these regulations seem to demand sound record keeping they do not mandate electronic record keeping or automated tracking and tracing. As a result, some food operators keep only paper-based documents. When a crisis or a scandal breaks out identification of the source of the problem takes much time and crisis response requires unnecessary effort from food operators that are not involved in the crisis.

A recent regulation, Reg. N^{o} 1169/2011 (EC 2011), on food labelling aims at providing consumers “a high level of health protection and to guarantee their right to information” so that they will have the right information “to make informed choices and to make safe use of food”. This regulation, which introduces additional obligatory nutritional labelling requirements, is also designed to keep up to date with consumers’ demand to new information and requires sufficient flexibility in transparency systems.

4.4.2 Business requirements

Whenever a crisis or a scandal breaks out the whole meat sector suffers. During the dioxin crisis of 1999 many companies in the meat as well as the feed sector suffered the consequences. In the recent horse meat scandal a large number of businesses (around 370) were affected and a large volume of meat (50,000 tonnes, part of which was already consumed) was recalled (Holligan 2013). Naturally, the vast majority of businesses in the meat sector would like to have a system in place that will overcome such crises or scandals from happening in the first place; they would naturally also like the response to a crisis or a scandal once it occurs to be surgical and quick.

Realizing a chain-wide transparency system for the entire meat sector is at present not realistic. Instead, it is possible to realize chain-wide transparency systems for specific supply chains as is the case in the illustrative example. However, the meat sector is characterized by many small businesses who can't afford to make large initial investments and implement state-of-the-art transparency systems. Even for medium and large food operators the cost-benefit analysis may not favour large investments because the break-even point of traceability may not be reached (Meuwissen *et al.* 2003). When transparency systems are in place as part of enterprise systems, detailed transparency data is rarely shared outside the boundary of the company (Trienekens *et al.* 2012) partly because transparency data are intertwined with sensitive business data (see section 4.4.4.3). These problems require solutions that will capture and process transparency data directly and that will accommodate a range of information systems, from simple ones used by small businesses to complex systems used by large food operators. 3pTSPs, defined here as trusted independent companies or organisations that provide transparency systems as a service, play a crucial role in facilitating information exchange.

4.4.3 Consumer requirements

Today consumers have only limited information about the products they buy and they have even more limited means of providing feedback. Since producers do not have effective means of communication with their consumers there arises a substantial communication gap between them and consumers (Duffy *et al.* 2005). In recent years smart devices (smartphones and tablets) enable to bridge this gap. There are today many examples of retail shops where instant and detailed product information is made available to consumers using smartphones (Ebling and Cáceres 2010). While such smartphone apps can be considered as a luxury convenience for many shoppers and consumers certain smartphone apps help improve the quality of life significantly. This is for instance demonstrated by the popularity of food allergy smartphone apps for sufferers of food allergy such as gluten intolerance.

However, even when a system like fTRACE is in place, communication between consumers and actors upstream the supply chain is in most cases not realized. For instance, individual farmers who have made animal welfare a priority have no way of informing consumers about their effort – and getting a better value for their product. Future transparency systems should reduce the communication gap between consumers and producers.

4.4.4 Technological requirements

New technological enablers provide new possibilities for realizing affordable and improved transparency systems. We identify standardization of transparency systems and new computing paradigms as two major technological enablers. These enablers impose requirements as technology push effects (Chau and Tam 2000) instead of pull effects (business, consumer or regulatory). Below we describe the effects of these enablers on future generation transparency systems in meat supply chains.

4.4.4.1 Standardization

Currently, transparency data in meat supply chains are mainly extracted from business transaction data, as the sector lacks widely adopted standards for capturing and communicating transparency data. But, business transactions do not contain all events that are relevant for transparency purposes. As a result relevant transparency information can escape detection. Moreover, the need to secure sensitive business data requires a solution that will capture and process transparency data independently from sensitive business transaction data (see section 4.4.2).

If such a solution is to be widely adopted it should be based on well-recognized standards. The EPCIS standard is such well-recognized global standard that can be used to realize internal and external transparency systems in the meat sector as demonstrated by few case studies (Grande and Vieira 2013, Hartley 2013). Below we describe what the standard specifies.

The EPCIS standard specifies how a trading company captures and stores data and makes it accessible to its trading partners. The standard adopts a distributed information system architecture. This means companies have full control over their data and provide and receive information using a common interface. Figure 4-5 depicts a transparency system based on the EPCIS standard.

The figure shows the components of a transparency system based on the EPCIS specification. The main components are EPCIS Capturing Applications, EPCIS Repositories, EPCIS Accessing Applications and Master Data Repositories. Capturing Applications are mostly linked to scanners, sensors or any other data capturing mechanisms. The data so captured are called EPCIS events. EPCIS events are also referred to as dynamic data because they are captured in the course of the product's journey through the supply chain and accumulate overtime. A Capturing Application sends EPCIS events to an EPCIS repository through the EPCIS Capture Interface of the repository. The repository makes data accessible through its EPCIS Query Interface. Trading partners query the repository using their EPCIS Accessing Applications. In this article we refer to the EPCIS repository and its capture and query interfaces as an *EPCIS system*. Master Data Repositories contain additional data (also referred to as static data) pertaining to products, locations of food operator facilities, or other contextual data that are necessary for describing EPCIS events (EPCglobal 2007).

The recent version of the standard (EPCIS 1. 1) defines four event types: an *object event*

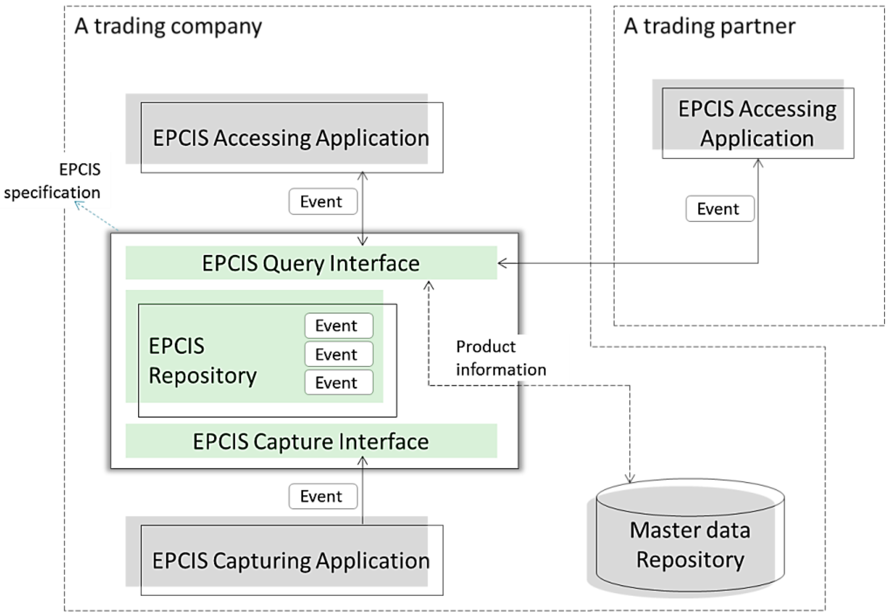


Figure 4-5. The components of an EPCIS-based transparency system and how information flows among the components. (adapted from EPCglobal 2007)

occurs when an object is observed (or is not observed while it should), an *aggregation event* occurs when an object is added to or removed from a containment (mainly used to track palletized objects), a *transaction event* occurs when an object is associated or disassociated with a business transaction, and *transformation event* occurs when one or more (input) objects are consumed and transformed into (output) objects. There is, in fact, a fifth event type, *quantity event*, coming from the previous version of the standard (EPCglobal 2007) used for counting the inventory level of a product. But, this event is deprecated in the new standard since the object event now enables to capture the same information what was previously captured by the quantity event. Events contain data about the identity of the product, the date and time of event occurrence, the location where it occurred, and the reason why the event occurred. These are conveniently abbreviated as the *what*, *when*, *where* and *why* of the event. The *what* (the identity) and *where* (location) are represented by EPCs and are globally unique. The *when* (date and time) is a local time and time zone or UTC (Universal Time Coordinated) timestamp. The *why* (reason) of the event is described using a predefined but extensible vocabulary of business process steps (GS1 EPCglobal 2014).

When users request for transparency information, traceability data from EPCIS repositories and contextual information from master data repositories are combined to create transparency information that is understandable to human users. Dynamic data become meaningful to users only when combined with static data (contextual information) that represent the various attributes of the events such as the description of the products, the locations where the events occurred and the circumstances of the events. For instance, consider a query that submits the ID of a product item (in the meat sector that could be the product code with a batch number) and gets a query result consisting of only one *object event* (in practice a query returns many events from more than one repository): "Object:*epc1*, Time*:t1*, Time zone: *z1*, ..., Location:*epcLoc1*, ...". To make this query result meaningful to users, the IDs *epc1*, *epLoc1*, *etc.* have to be translated to meaningful information, therefore, *epc1* and *epcLoc1* have to be looked up in a master data repository. The look up may, for instance, return "veal shoulder blade, *etc.*" for *epc1* describing the product and "*x* house number, *y* street, *z* city, *etc.*" for *epcLoc1* describing the full address of the processing plant where the event is generated (EPCglobal 2007).

Querying for transparency information is a challenge since the product information has to come from several EPCIS repositories whose web-addresses have to be discovered. Specifically, given an EPC (an ID of a specific product item) one should be able to obtain pointers to the EPCIS systems from which one can retrieve the events related to the EPC. Two standards are proposed for this purpose. The ONS (Object Naming Service) standard (EPCglobal 2013) is a standard that defines how a product EPC can be resolved to the authoritative EPCIS system associated with the issuer of the EPC. The Discovery Services standard, which is still work-in-progress, will enable to discover who else may have information for a given EPC (GS1 EPCglobal 2013).

The fact that the discovery services standard is still under development while almost all other EPC related standards were made available in quick succession between 2004 and 2007 and the few remaining by 2011 (GS1 EPCglobal 2013) indicates the difficulty of devising a way of global discovery. We argue instead for supply chain specific discovery service provided by 3pTSPs. Since 3pTSPs serve specific supply chains (see section 4.4.2) they can also provide integration with legacy (non-EPCIS) transparency systems. Integration with legacy systems is crucial in realizing chain-wide transparency since some food operators have already invested on these legacy systems, and these food operators may not be willing to adopt yet new transparency systems. In addition, end-user applications (such as smart phone apps for consumers) are preferably made available by 3pTSPs instead of by individual food operators for diverse reasons including reducing the cost of development of the applications and promoting their widespread use.

4.4.4.2 New computing paradigms

Realizing a transparency system requires a set of software systems that may require large initial capital investments in software and hardware. Cloud-computing enables software to be available as a utility or service (instead of a product) making large initial capital investments in software and hardware unnecessary. Cloud computing refers to software applications, or the lower-level infrastructure for building software applications, delivered as services or over the Internet (Armbrust *et al.* 2010). Since the cloud paradigm to computing is relatively recent, no coherent set of generic functionalities were available. In Europe the FI-PPP program aims to change this by providing the FI-Ware platform and the resulting cloud-based platforms and software applications.

The FI-Ware platform offers GEs for building new software platforms and applications. There are currently dozens of GEs available in the FI-Ware platform. These GEs are categorized as Cloud Hosting (CH), Data and Context Management (DCM), Internet of Things (IoT), Interface to Networks and Devices (IND), Identity and Security (I&S) and Marketplace and Mashup Frameworks (MMF) GEs (FI-Ware 2013). FIspace is one of the platforms built using FI-Ware GEs. FIspace is a business collaboration platform designed as a ‘social-network’, much like Facebook, but for businesses. The platform provides a set of basic applications, a business process engine that allows systems designers and software developers to link the apps and services into a collaboration business process, and a set of APIs for developing apps and business processes. The development of FIspace as a SaaS platform fits the current trend that is characterized by increasing adoption of the cloud computing paradigm (Patidar *et al.* 2012).

4.4.4.3 Constraints on technical requirements

Functional requirements are usually accompanied by non-functional characteristics such as usability, security, performance, *etc*. (Glinz 2007, Chung and Leite 2009). Many of these non-functional characteristics (referred to as non-functional requirements) specify implementation or external constraints (Glinz 2007) and thus depend on the technical choices made in a specific software product development rather than the choices made in a reference architecture. But, some non-functional requirements may directly be related to the choices made in the reference architecture. Specific non-functional requirements put forward during the pilot study are mainly related to data security, namely: data ownership, data storage location and vendor (in)dependence. Many businesses want to maintain ownership of their data so that they can control who may or may not have access to them. They want to have control over where their data will be stored; they want to decide on which data will be stored locally and which in *the cloud* (remote data repositories). Many businesses demand that they be able to specify the confidentiality of their data that are stored remotely and accessed through cloud-based services. In addition, businesses want to avoid getting trapped in vendor specific platform. In case of FIspace they made their preference for multi-instance and distributed FIspace platform over a single instance (like Facebook) platform.

4.5 Results and Discussions

A software architecture describes the components of a software system, their interactions and the relation of the system with its environment based on software design principles, styles and patterns (Bass *et al.* 2003, ISO/IEC/IEEE 2011). A reference architecture describes these characteristics but for a class of software systems using a combination of one or more reference models and architectural patterns (Bass *et al.* 2003 pp. 24-6). In this section we describe our proposed reference architecture and show how it addresses the generic requirements outlined in the previous section.

4.5.1 Transparency system based on EPCIS

By choosing for the EPCIS standard to fulfil one of the generic requirements (see section 4.4.4.1) we also choose to align the way data is captured, stored and accessed with what is specified in the standard. Since the EPCIS standard is not specifically made for, and does not fully address the requirements of, meat supply chains we shall identify where it meets our needs, what its shortcomings are, and how the shortcomings can be addressed.

According to the EPCIS standard transparency data are captured as events unfold during the physical flow of products. Events are captured with the help of four data items: *what* (in the meat sector using a Global Trade Identification Number (GTIN) (GS1 2013) together with the batch ID), *where* (usually using a Global Location Number (GLN) (GS1 2013)), *when* (using a globally unique timestamp (ISO 2004)), and *why* (using standard vocabularies). Capturing meat transparency data as EPCIS events requires understanding the processes involved in meat supply chains and identifying where and when an event should be captured. At the farm level, events critical for transparency are the birth of animals and their transfer from one farm to another farm or to a slaughterhouse. Other relevant events include those related to feed, vaccination, medication and veterinary inspection. At processing facilities critical events are the slaughter of the animal, the processing of meat (splitting, chilling, cutting/portioning, *etc.*), inspection, testing and packaging. In logistic operations critical events are transport (loading and unloading) and storage. At retail the important event is the sale of the product. At all stages the withdrawal of a product for whatever reason is an important event that needs to be captured.

To address specific aspects of the meat supply chains we identify two aspects related to metadata where transparency systems – even if built to the EPCIS standard – need to be augmented. First, metadata vocabularies related to business operations in the meat sector need to be defined by the sector. While the vocabularies used for logistic operations, such as *loading*, *picking*, *packing*, *etc.*, are common across many sectors (Blackstone and Cox 2005, GS1 EPCglobal 2014), meat production processes like *birth*, *medication*, *slaughtering*, *splitting*, *etc.* have to be defined in a uniform manner.

Second, meat product metadata that provide consumer awareness about meat product quality attributes such as nutritional value, ingredients, and safety should be captured in product master data repositories. Various meat attributes, such as cuts and grades, are standardized to facilitate communication and electronic trading (UN/ECE 2006, Polkinghorne and Thompson

2010). However, such information is also valuable to consumers and should be part of the transparency information provided to them.

4.5.2 Storing and sharing transparency data

As the EPCIS standard addresses the requirement for electronic record keeping and data sharing, it also requires that all food operators implement an EPCIS software system within their organisations. However, small companies in the meat sector (mainly farmers) use basic information systems or simple desktop applications. They deal with limited amount of transparency data that, in many cases, will not justify implementing full-fledged EPCIS systems (see section 4.4.2). We argued for 3pTSPs that will fill this gap. 3pTSPs can fill this gap by providing EPCIS systems that can be either shared among several companies or leased privately, in both cases based on the *use on-demand, pay-as-you-go* (Armbrust *et al.* 2010) business model to IT provisioning.

Figure 4-6 shows how food operators, large and small, can use the services provided by 3pTSPs. Small businesses that do not have their own EPCIS system use a shared EPCIS system to store and share transparency data. 3pTSPs provide them with a web-based interface for uploading and managing data **(1)**. Large food operators with their own EPCIS systems provide EPCIS query interfaces with which others can query for transparency data **(2)**. Sharing compulsory transparency information with immediate suppliers and customers to fulfil regulatory requirements (see section 4.4.1) can be realized using either the shared EPCIS system or direct exchange of information through mutually agreed protocols **(3)**. If some of the companies use proprietary protocols or other standards (such as IsoAgriNet) 3pTSPs can adopt the additional methods in **(1).**

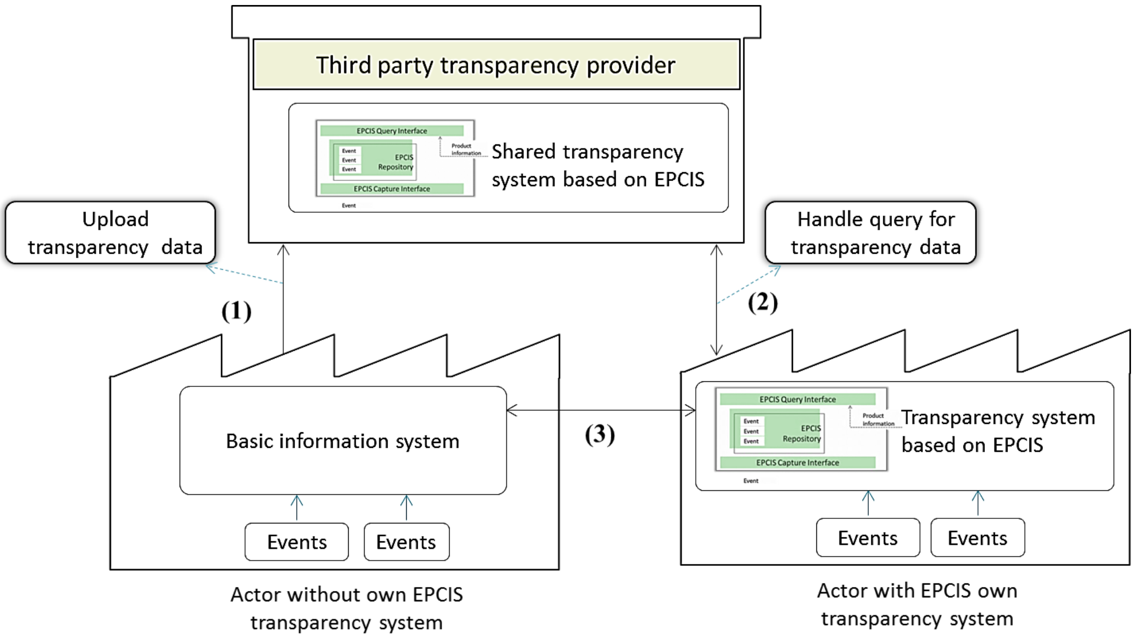


Figure 4-6. Sharing transparency data. (1) Transferring transparency data to a shared EPCIS repository managed by a 3pTSP. (2) Querying transparency data using query interfaces. (3) Sharing compulsory transparency information with immediate suppliers and customers directly using either EPCIS query interfaces or through mutually agreed protocols.

4.5.3 *Transparency system powered by cloud-based services*

We envisage that EPCIS systems are better provided as cloud-based services for at least three reasons. First, small businesses may only afford a shared EPCIS system provided as a cloud-based service by 3pTSPs. Second, not all food operators who want to use their own EPCIS system may choose for a potentially expensive on-premise EPCIS system; they would rather use a cloud-based EPCIS system (see section 4.4.4.2) leased from 3pTSPs. Third, the necessary query and discovery services are more easily made available as cloud-based services provided by 3pTSPs. In the remainder of this section we describe a usage scenario of a single shared EPCIS system; scenarios involving more than one EPCIS system (shared, on-premise or cloud-based) are described in section 4.5.4.

Figure 4-7 depicts a shared EPCIS system, provided by a 3pTSP, and its relationship with its environment – *i.e.* the FIspace collaboration platform (and through which the FI-Ware GEs), supply chain companies, and end-users (members of the 3pTSP, participating companies, authorities and consumers). The FIspace platform provides basic apps (some of which are recently referred to as services) which do one specific job. The apps support business processes by providing the logic and the user interface for specific task they are developed for while the business process engine (part of the core internal feature of FIspace – not depicted in the figure) manages the status of the tasks to be done.

The five apps depicted in the picture: *business profile app*, *product information app*, *logistic planning app*, *marketplace* app and *SLA (Service Level Agreement) management app*, represent the basic features of the platform. A business profile app is used by businesses to

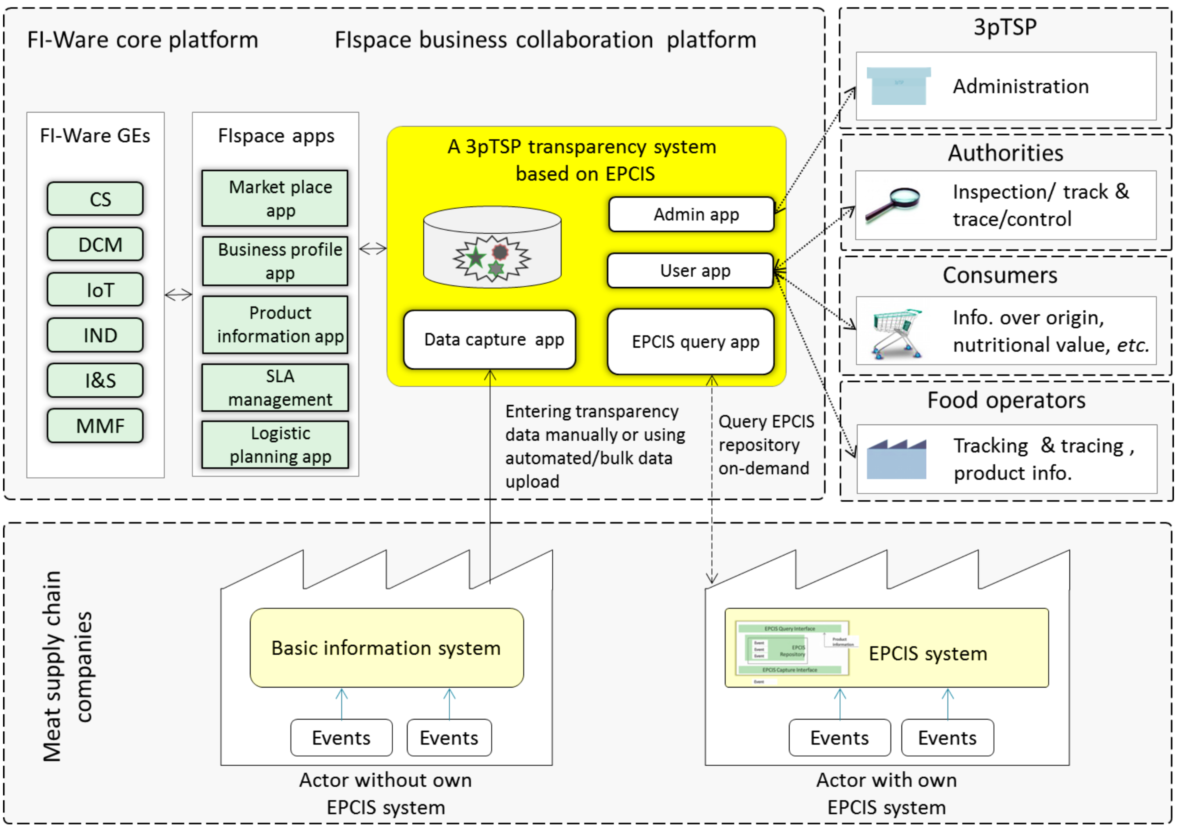


Figure 4-7. A chain-wide transparency system based on FIspace: the various apps provided by 3pTSPs and their users

maintain their business profiles in FIspace. A product information app is used to display information about products and services. A marketplace app is used to announce products and services businesses are interested in or are offering. A logistic planning app enables businesses to plan transport using real time logistic information. An SLA management app provides services related to SLAs such as establishing SLAs and signalling deviation from them during execution.

3pTSPs can leverage these basic apps and most importantly provide new ones required for realizing chain-wide transparency systems. We identify four new apps that 3pTSPs should provide: *admin app*, *user/consumer app*, *data capture app* and *EPCIS query app*. The admin app is a type of SLA management app that allows the 3pTSP and its customers (food operators) to manage agreements concerning data sharing and other transparency related services. These include, among others, managing subscription to systems provided by the 3pTSP and managing data access rights to users. The other three apps are variations of a product information app. The user/consumer app resembles the fTRACE smartphone app (see Figure 4-3 and section 4.4.3) and provides users access to transparency information. The capture and query apps are used to capture and query EPCIS event data. In EPCIS terms the capture app is an EPCIS Capturing Application; likewise the query app is an EPCIS Accessing Application. The user app uses query results from the query app to provide consumers with user-friendly information.

4.5.4 Chain-wide transparency system usage scenarios

Figure 4-8 shows how a chain-wide transparency system involving a number of food operators can be set-up. The FIspace platform simplifies the setting up of the transparency system because the platform provides food operators basic services that enable them to present their business profiles, discover trading partners, view product offerings and facilitate business interactions. To be part of such a chain-wide transparency system food operators should use one of the three options: implement their own internal transparency system based on the EPCIS standard, use a shared EPCIS system in FIspace offered by a 3pTSP or adapt their legacy internal transparency system in a way that it can be queried by the apps of

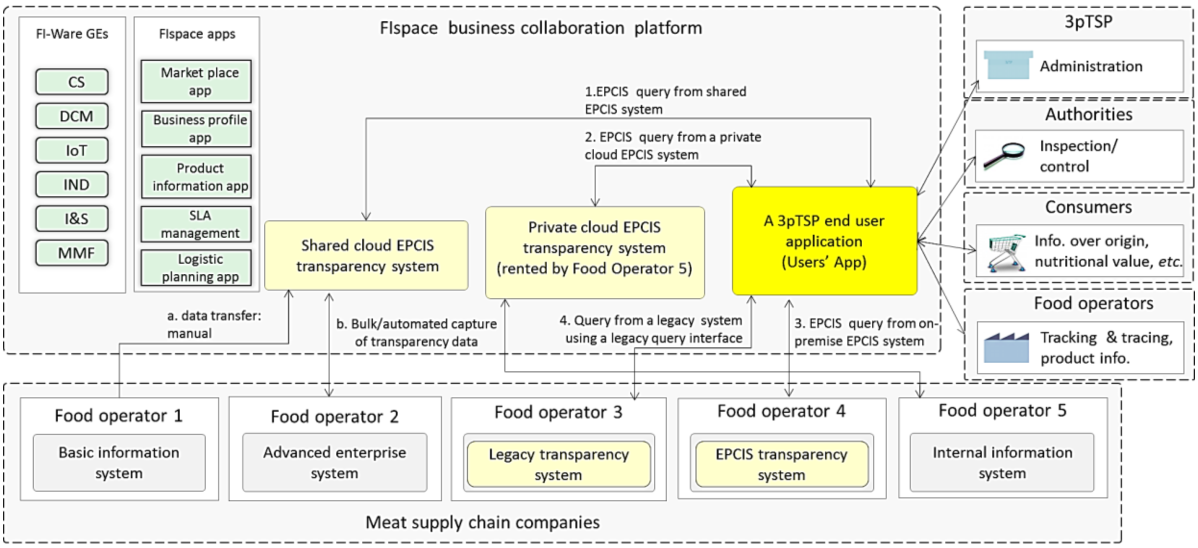


Figure 4-8. A chain-wide transparency system based on FIspace: how various types of food operators provide access to transparency information

3pTSPs (see sections 4.4.2 and 4.4.4.3). Food operators using a shared EPCIS system can choose between two ways of transferring transparency data to the shared system: a) manual data transfer, or b) automated data capture. We identify four types of data query with the corresponding internal transparency system: 1) EPCIS query from a shared EPCIS system, 2) EPCIS query from a cloud-based EPCIS system, 3) EPCIS query from on-premise EPCIS system and 4) proprietary query from a legacy system.

The various types of food operators that can take part in a chain-wide transparency system provide us with three major scenarios of chain-wide transparency. First, we consider the two extremes and the corresponding scenarios, which are less likely to be used in practice. Next, we elaborate the most likely scenario. In all scenarios we assume there is a 3pTSP either established by the trading food operators (as a consortium) or as independent business as is the case in the fTRACE system. The 3pTSP provides a shared EPCIS system instance fully managed by itself. It also leases EPCIS system instances to food operators who will use and manage them privately. On-premise EPCIS and non-EPCIS transparency systems reside outside of the FIspace platform.

On the one extreme all food operators will have their own EPCIS systems (*i.e.* all food operators are of type 4 or 5 in Figure 4-8). Food operator 5 differs from Food operator 4 in that Food operator 5 uses a cloud-based EPCIS system offered by a 3pTSP. Food operator 4 and 5 type companies are usually medium and large companies. Since all food operators have their own EPCIS systems a shared EPCIS system is unnecessary. In this scenario all food operators have also full control over their transparency data and can control who accesses what type of information at all times. The main purpose of the 3pTSP is providing a discovery service and a consumer app, which the various customers of the food operators can use to get transparency information.

On the other extreme all food operators use a shared repository (*i.e.* all food operators are of type 1 or 2 in Figure 4-8). Food operator 1 represents small companies (*e.g.* small farmers) who use basic ICT systems and generate only a limited amount of transparency data that are usually recorded manually. This food operator will use a data capture app to manually enter transparency data in the shared EPCIS system. Food operator 2 represents medium-sized companies that use advanced information systems but no internal transparency systems. This food operator generates relatively large amounts of transparency data and the preferred mechanism of transferring data to the shared transparency system is through an automated bulk data upload. This food operator will use the capture app through the data capture API. In this scenario complying with the EPCIS standard may not even be necessary because this scenario represents a centralized system. This scenario is similar to the fTRACE system used in our pilot study supply chain.

A practical scenario that lies between the two extremes accommodates the four different types of food operators that use EPCIS systems. In this scenario a food operator may choose to use either a shared EPCIS system or its own private (on premise or cloud-based) EPCIS system. This guaranties that all nodes of the distributed system are EPCIS systems as is required by the EPCIS standard.

Besides the four types of food operators, meat supply chains usually involve yet another type of food operator: Food operator 3 (see Figure 4-8). This food operator represents most of the large food operators in today’s meat supply chains. This food operator has its own non-standard (thus non-EPCIS) internal transparency system and cannot directly exchange information with those who use EPCIS systems. This food operator may not be willing to upgrade its system to adopt the EPCIS standard and, as a result, creates the greatest challenge in realizing a chain-wide transparency system. For a 3pTSP to serve a supply chain with this type of food operators the 3pTSP should support the proprietary query interfaces and data formats the food operator uses.

To provide chain-wide transparency involving EPCIS and legacy (non- EPCIS) systems requires a tedious work of coupling. Therefore, 3pTSPs will play a crucial role in devising practical solutions for the current day situations. However, to be able to join future chain-wide transparency systems food operators of type 3 may have to use private internal EPCIS systems and become a type 4 or 5 food operator or use a shared transparency system and become a type 1 or 2 food operator.

4.6 CONCLUSION

In this paper we presented a reference software architecture for chain-wide transparency systems in meat supply chains. We showed how such transparency systems can be realized based on the EPCIS standard and using cloud-based services. We argued for third-party transparency service providers who will play an important role of providing shared and private EPCIS systems as cloud-based services, integration with legacy transparency systems, end-user apps and a discovery service to identify on-premise EPCIS systems.

Before designs based on the reference architecture can be widely implemented and evaluated in practice the proposed architecture should be tested. Currently a prototype chain-wide transparency system and the platform on which it will run (the FIspace collaboration platform) are being developed. When these are completed, we will be able to start evaluation in practice.

ACKNOWLEDGEMENT - The research leading to this paper received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) within the FI-PPP’s project SAF under grant agreement n° 285 326 and project FIspace under grant agreement n° 604 123. The authors acknowledge the individuals and companies involved for their support and the European Community for their funding. We would also like to acknowledge Angela Schillings-Schmitz of GS1 Germany for conducting the interviews and Sabine Kläser from the same organisation for inputs on relevant meat regulations.

5

Realizing Chain-Wide Transparency in Meat Supply Chains using a Reference Architecture[1]

[1] Published in the journal of Computers and Electronics in Agriculture (ISSN: 0168-1699): Kassahun, A., R. J. M. Hartog and B. Tekinerdogan (2016). Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture. *Computers and Electronics in Agriculture*. **123**: 275-291.

Abstract

One of the key concerns in meat supply chains is to provide chain-wide transparency, whereby food operators capture and share transparency data across the supply chain. To meet this concern a chain-wide transparency software system is needed that is able to address the desired stakeholder requirements. Unfortunately, designing and implementing a chain-wide transparency system is not straightforward. In this paper we provide a systematic approach for designing and implementing chain wide transparency systems. To this end, we first present a reference architecture that represents a generic design of such systems. Secondly, we discuss the systematic approach for deriving concrete architectures from the reference architecture based on stakeholders' requirements. Finally, we illustrate our approach by designing and implementing a transparency system for beef supply chains.

Keywords: meat supply chains; chain-wide transparency; transparency systems; EPCIS; reference architecture.

5.1 INTRODUCTION

Lack of transparency in meat supply chains is a major problem, which has become evident from recurring crises and scandals involving meat products. A chain-wide transparency system will enable food operators to manage transparency data within their facilities (*i.e.* internal transparency) and to share transparency data with other food operators and stakeholders (*i.e.* external transparency) (Moe 1998, Bertolini *et al.* 2006, Gandino *et al.* 2009, Bosona and Gebresenbet 2013). When suitable internal and external transparency systems are not in place chain-wide transparency fails.

Internal transparency requires capturing the events that take place within the food operators. In the meat sector these events concern the things that happen to animals (such as birth, feeding, treatment, movement and slaughtering), and meat (such as splitting, cutting, mixing, transport and storage). To realize external transparency all food operators across the entire supply chain should use an internal transparency system and in addition these systems should comply with common standards for sharing transparency data.

In literature various chain-wide transparency systems are proposed. The most influential are food regulations which all food operators are required by law to comply with. In Europe, for instance, food operators must comply with the General Food Law (EC 2002), regulations on mandatory registration of animals (EC 2000, EC 2004, EC 2015), and regulations for tracking and tracing of meat products (EC 2007, EC 2011). The state of chain-wide transparency in European meat supply chains is currently largely determined by these regulatory mandates. Regulations, however, do not specify how the systems have to be realized. Moreover, regulatory requirements are not strict enough to cover the needs of all stakeholders and mandate greater level of transparency. As a result current chain-wide transparency systems in place are not adequate (Kassahun *et al.* 2014).

Several researchers addressed the shortcomings of current transparency systems, often focusing on parts of the larger puzzle. Some focused on farms and proposed transparency systems for capturing and sharing transparency data about animals beyond what is mandated by regulations, such as data on what the animals are fed and when and how they are treated (Shanahan *et al.* 2009, Voulodimos *et al.* 2010). Others focused on meat processing facilities and showed how a meat product can be tracked as it undergoes various transformations (cutting and mixing) during meat processing (Mousavi *et al.* 2005, Donnelly *et al.* 2009). Still others focussed on the sharing of transparency data and demonstrated how transparency standards can be used to address this aspect of chain-wide transparency (Shanahan *et al.* 2009, Thakur *et al.* 2011, Feng *et al.* 2013). Although several meat supply transparency systems have been proposed, designing and implementing a chain-wide transparency system for a particular meat supply chain remains a difficult problem.

A common solution for addressing this problem is the use of *reference architectures* (Cloutier *et al.* 2010, Angelov *et al.* 2012). A reference architecture is a generic design that assists architects to derive *concrete architectures* for particular contexts. In this paper we present a reference architecture for chain-wide transparency systems. Depending on stakeholder requirements the reference architecture can be used to derive different alternative concrete architectures. However, deriving a concrete architecture involves many different design decisions and likewise it is not easy to derive a feasible architecture. Moreover, once a concrete architecture has been derived implementing the system based on the architecture is far from trivial.

In this paper we provide a systematic approach to support the design and implementation of chain-wide transparency systems. To this end, we first present a reference architecture that represents a generic design of such systems. Secondly, we discuss the systematic approach for deriving concrete architectures from the reference architecture based on stakeholders' requirements. Finally, we illustrate our approach by designing and implementing a Chain-wide Meat Transparency System (CMTS) for beef supply chains.

The rest of the paper is organised as follows. Section 5.2 provides background information about the reference architecture. Section 5.3 summarizes related work and provides the problem statement. Section 5.4 describes the process for deriving a concrete architecture from the presented reference architecture. Section 5.5 presents CMTS demonstrating how its architecture is instantiated. In section 5.6 we conclude the paper.

5.2 Background

5.2.1 Software Architecture

Every software system has a software architecture that defines its design. This is not different for a chain-wide transparency system. A software architecture describes the components of the system, the interactions among the components, and the interaction of the system as a whole with its environment (ISO/IEC/IEEE 2011, Bass *et al.* 2012, Tekinerdogan 2014). A software architecture is an abstract representation that identifies the gross-level structure of the system and is important for supporting the communication among stakeholders, for guiding the design decisions, and for analysis of the overall system (Tekinerdogan 2014).

A software architecture that addresses the concerns of specific stakeholders is here referred to as concrete architecture. Hereby, a stakeholder is defined as an individual, team, or organisation with interests in, or concerns relative to, the system. A concrete architecture defines the boundaries and constraints for the implementation and is used to analyse risks, balance trade-offs, plan the implementation project and allocate tasks (Tekinerdogan 2014).

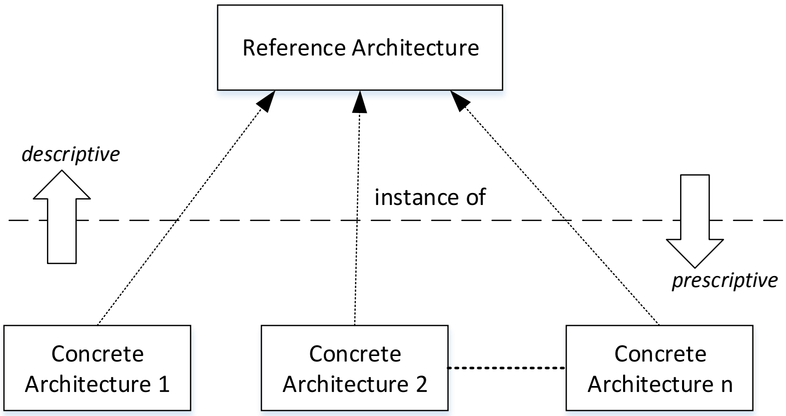


Figure 5-1. Relation between reference architecture and concrete architectures

Concrete architectures can be viewed as instances of a reference architecture, which is a generic design. In turn, a reference architecture is derived from the knowledge and experiences accumulated in designing concrete architectures in the past (Cloutier *et al.* 2010, Angelov *et al.* 2012). The concrete architectures differ from one case to the next depending on the requirements of the stakeholders involved. Reference architectures can be used descriptively to "capture the essence of existing architectures" or prescriptively to guide the development of new ones (Cloutier *et al.* 2010). Figure 5-1 depicts the relations between reference architecture and concrete architectures.

5.2.2 A reference architecture for chain-wide transparency systems

We have provided an initial architecture for chain-wide transparency systems in an earlier study (Kassahun *et al.* 2014) to discuss the different concerns in meat supply chains. Figure 5-2 depicts the complete reference architecture that elaborates on this earlier work. The figure is described in detail in the following sub-sections.

5.2.2.1 Stakeholders

The reference architecture distinguishes between three main types of stakeholders, *food operators* (*fo*), *end-users* (*eu*), and *third-party* (*3p*) service providers. Food operators provide transparency data about their products and operations. End-users are individuals and organisations who wish to access transparency data. Third-party service providers facilitate chain-wide transparency by providing and managing transparency software systems. The stakeholders are summarized in Table 5-1.

Food operators

Five types of food operators are identified based on whether or not they have an internal transparency system in place and its types. We identify two types of food operators who do not have their own private transparency system. They contribute to chain-wide transparency by transferring transparency data to the shared repository, where the data will be stored and shared. Small food operators, such as farmers, who use basic IT systems, enter data manually through the web interface of the shared repository; and they are labelled as *type 1 food operator* (fo_1). Large food operators, such as slaughterhouses, who have advanced IT systems in place, will most likely use automated batch data transfer, and they are labelled as *type 2 food operator* (fo_2).

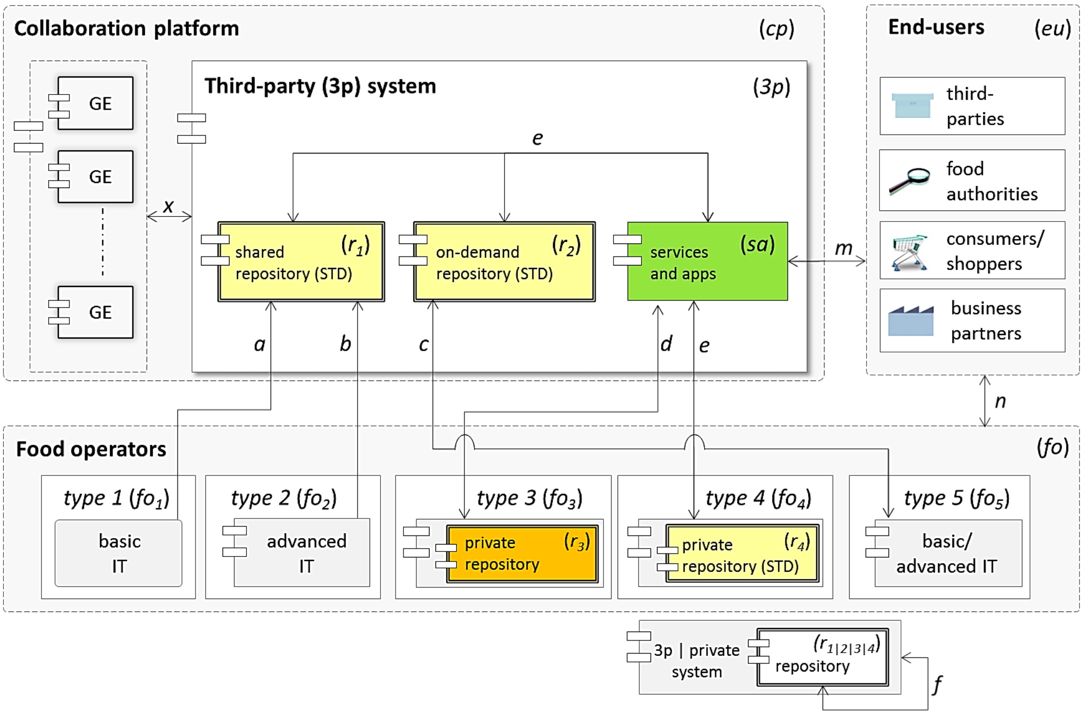


Figure 5-2. Reference architecture for chain-wide transparency systems for the meat sector (adapted from Kassahun et al. 2014)

We further identify three types of food operators who have a private transparency system. These systems are considered part of the chain-wide transparency system. Food operators who use a *legacy* transparency system (see next section for the definition of legacy) are labelled as *type 3 food operator* (*fo*$_3$); and those that use a standards-compliant (STD) transparency system are labelled as *type 4* (*fo*$_4$) or *type 5* (*fo*$_5$) depending on where the system is deployed. Food operators who deploy and manage their own transparency systems are *fo*$_4$; those who use on demand transparency systems following a cloud business model are *fo*$_5$.

End users

We can identify four categories of end-users: *consumer/shopper*, *business partner*, *food authority*, and *third-party*. Consumers and shoppers are individuals who mainly want to know more about the meat products they buy or consume. Business partners are the business customers and associates of the food operators, including the food operators of the supply chain. They need access to transparency data as part of their business dealings. Food authorities are legal authorities who need to, and are mandated, to access transparency data. Such is the case, for instance, during food alerts. Third-parties are those who provide transparency, certification or accreditation services.

Service providers

Besides food operators and end-users, the reference architecture identifies transparency service providers called *third-party Transparency Service Providers* (3pTSPs). They provide and manage the shared and private transparency systems that are used on-demand.

Table 5-1. The main stakeholders of a chain-wide transparency system

Stakeholders	Description
Food operators	
Type 1 food operator (fo_1)	Food operators who share transparency data through the shared repository. They enter transparency data into the shared repository manually.
Type 2 food operator (fo_2)	Food operators who share transparency data as *type 1* food operators but use automated process to transfer transparency data into the shared repository.
Type 3 food operator (fo_3)	Food operators who use a legacy private transparency system.
Type 4 food operator (fo_4)	Food operators who use private transparency system that comply with standards chosen by the 3pTSP.
Type 5 food operator (fo_5)	Food operators who use on-demand transparency system that is hired from the 3pTSP.
End-users	
Consumer/Shopper	Individuals interested in transparency data about a meat product they buy or consume.
Business partner	The business partners of the food operators that need access to transparency data as part of their business transaction.
Food authority	Regulatory authorities who are mandated to access transparency data in case of, for instance, food alerts.
Third-parties	Independent third-parties that facilitate chain-wide transparency by providing systems, inspections, certifications, *etc.*
System providers	
3pTSP	A third-party transparency service provider, who is one of the independent third-parties, that provides and manages the shared transparency system.

5.2.2.2 Components

The components of the reference architecture can be grouped into two categories: the components of a shared transparency system provided by a 3pTSP (*3p system*) and the components of a set of distributed private internal transparency systems (*private systems*) of the food operators. The components can be arranged following a *centralized*, a *distributed* or a *hybrid* architecture. Thus, an instantiation of the reference architecture may constitute exclusively of the components of the 3p system, exclusively of the components of the private systems, or a combination of both. Table 5-2 summarizes the components of the reference architecture.

The 3p system consists of *repositories* (r_1, r_2), and *services and apps* (*sa*). Likewise, the private systems contain a repository and a set of apps. To realize chain-wide transparency both the 3p and private systems should adopt common transparency standards. In a centralized or a hybrid architecture the 3pTSP defines or selected the standards that will be used across the supply chain. In a distributed architecture the food operators have to agree on a common set of standards. In all cases the use of global standards is preferred. The standards adopted should at least prescribe the format of transparency data and interfaces for capturing and querying the repositories of the 3p and private systems. A private system that does not comply with the common standards is considered as a legacy (proprietary) system. As the 3pTSP selects or defines the standards, the 3p system is assumed to always comply with the standards. Since the 3p system provides the desired set of apps the private systems may consist only a repository but no apps. If a food operator includes app in its private system, then the apps are used only by its customers and business partners. However, the data in the repository of the private system is shared. Therefore, hereafter we use the term *private system* mainly to refer to the repository of the private system, and we use the terms *private system* and *private repository* interchangeably. The *3p services* facilitate interaction between the apps (3p or private apps) and the repositories. The apps provide external systems (used by end-users or food operators) access to the repositories.

Table 5-2. The main components of a chain-wide transparency system

Component	Description
Shared system	
Share repository (r_1)	A shared and standard-compliant repository that is part of the 3p system.
Service (sa)	A functionality of the 3p system that is accessed through an Application Programmers Interface (API).
App (sa)	An interactive software application (app), with a Graphical User Interface (GUI), that is part of the 3p system.
Private systems	
Legacy private repository (r_3)	A private on-premise repository that does not comply with the common standards.
Standard private repository (r_4)	A private on-premise repository that does comply with the common standards.
Standard cloud repository (r_5)	A repository offered by the 3pTSP to food operators to be used on-demand privately.
Collaboration platform	
Generic Enabler (GE)	A set of generic enablers provided by the underlying platform.

The private systems contain either on-demand or on-premise repository. The term on-demand repository (r_2) refers to a transparency data repository offered by the 3pTSP following a use-on-demand, pay-as-you-go cloud business model. On-premise repository (r_3 or r_4) are deployed and operated by the food operators themselves. If the private system complies with the standards, then its repository is standard-compliant (r_4), otherwise it is a legacy repository (r_3). A legacy repository is part of the chain-wide transparency system as long as its transparency data can be accessed – either using the 3p apps or the underlying platform (*cp*).

The 3p system, like any software system, is designed for, and deployed in, a particular platform. These platforms are increasingly available as utility following a cloud computing paradigm and provide a diverse set of generic enablers (GEs) (Castrucci *et al.* 2011). The realization of the 3p system can be facilitated by using GEs) or collaboration platforms such as FIspace (*cp*).

5.2.2.3 Interactions

The reference architecture identifies various linkages representing the interactions either among the components of the chain-wide transparency system or the system as a whole with external systems. These interactions are grouped into three types of interactions: *data capture, internal data query* and *external data query* (see Table 5-3).

Data capture

Data capture links represent the interactions between food operators and the repositories of the 3p system. The link (*a*) represents manual data entry over the internet by a type 1 food operator through an interactive GUI. The link (*b*) represents automated batch data transfer through an API using an advanced IT system by a type 2 food operator. Both interactions link the food operators with the shared repository. In the future food operators may use a cloud-based on-demand repository rented from the 3pTSP. The link (*c*) represents a standard API call through which real-time transparency data is transferred from the facilities of a type 5 food operator to its rented repository managed by the 3pTSP.

Data query: internal

Internal data query links represent queries for transparency data within a chain-wide transparency system. The link (*d*) represents a non-standard interface between the 3p system and the legacy system of a type 3 food operator; the link (*e*) represents a standard interface between the 3p system and a standard-compliant private system of a type 4 or a type 5 food operator. The interactions can be initiated either by the 3p system or by the private systems. Often, it is the 3p apps that initiate an interaction for the purpose of querying transparency data. However, in a distributed architecture, the apps of the private systems may initiate interaction with the 3p system to discover the other repositories of the chain-wide transparency system. The link (*f*) represents interfaces between two private systems. If both private systems are standard-compliant data queries along the (*e*) and (*f*) are standard queries; otherwise, they are non-standard queries.

Data query: external

End-users access transparency data through 3p apps (*m*). In some cases customers and business partners of food operators may use the private apps of the food operators to access transparency data (*n*). Besides, the 3p system may use the GEs of the underlying collaboration platform (*x*) for some core functionalities such as security and interfaces to legacy systems (Moltchanov and Rodriguez Rocha 2014).

Table 5-3. The main interactions involving the components of a chain-wide transparency system

Link	Description
Data capture	
$fo_1 - r_1$ (*a*)	Manual data entry into the shared repository using a GUI.
$fo_1 - r_1$ (*b*)	Automated batch data transfer into the shared repository..
$fo_5 - r_2$ (*c*)	Automated real-time data capture by cloud-based private repository.
Data query: internal	
$sa - r_3$ (*d*)	Non-standard data query between 3p services and apps and a legacy system.
$sa - r_{1\|2\|4}$ (*e*)	Standard data query between 3p services and apps and standard-compliant repositories.
$r_{1\|2\|3\|4} - r_{1\|2\|3\|4}$ (*f*)	Direct data query between two repositories.
Data query: external	
eu – 3p (*m*)	End-users using 3p apps.
eu – fo (*n*)	End-users using apps provided by the food operators.
3p – cp (*x*)	The 3p system using the GEs of the underlying collaboration platform.

5.3 RELATED WORK AND PROBLEM STATEMENT

Using the reference architecture that we have described in the previous section we can describe existing concrete architectures. In this section we will use the reference architecture to discuss the related work on concrete architectures of transparency systems for meat supply chains in section 5.3.1 to section 5.3.3. Further, based on this discussion we will formulate the problem statement that we address in this paper in section 5.3.4.

5.3.1 An architecture following a centralized approach

For the European meat sector the European food regulations (EC 2000, EC 2004, EC 2015) require member states to setup a national animal registration system in which bovine, and recently equine, animals are registered. These regulations effectively demand a centralized approach in which the transparency data is located on a centralized server managed by third-parties (which are the national food authorities of member states). Food operators report the birth, the movement and the death of each individual animal to their national registration system. Obviously, the scope of these systems is limited to transparency about animals and, therefore, it only affects farm (mostly *type 1*) and slaughterhouse (mostly *type 2*) food operators. The architecture for these systems is depicted in Figure 5-3.

Some researchers built on this architecture to provide greater transparency about animals. Voulodimos *et. al.* (2010) demonstrated how local databases at farms can be used in conjunction with a central repository to gather and share detailed transparency data. For that purpose they used RFID (Radio-Frequency IDentification) tags attached to animals to gather data, such as, the movement and feeding pattern of the animals. This approach will enable supply chains to realize full chain-wide transparency if combined with greater transparency in meat processing plants where each carcass is tracked (see, for instance, Mousavi *et al.* 2005).

5.3.2 An architecture following the one-step-back/one-step-forward principle

An architecture following the *one-step-back/one-step-forward* principle follows naturally from needs of business transactions. Businesses naturally keep administration of their supplies and sales. However, European food regulations formally require food operators to be able to identify their suppliers and customers (except consumers) for transparency purposes (EC 2007, EC 2011). In principle, the *one-step-back/one-step-forward* principle applies to all food

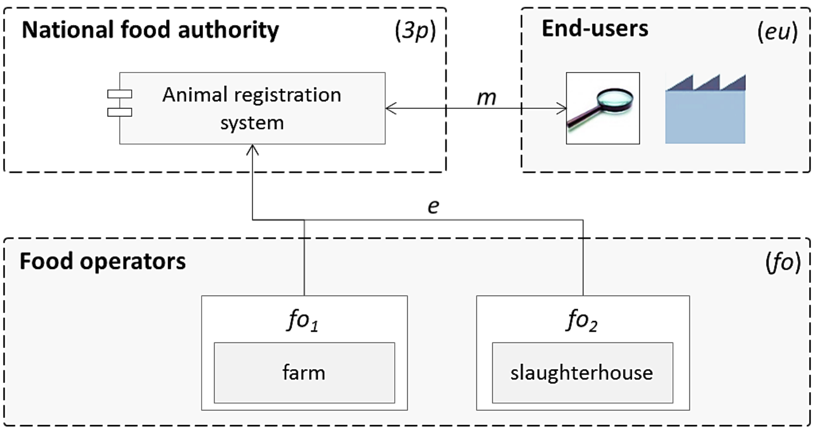


Figure 5-3. An architecture following a centralized approach as an instance of the reference architecture

operators in the meat sector. However, this regulation is more relevant for slaughterhouses and other meat processing plants that are not required to use a centralized registration system.

Figure 5-4 presents the architecture which adopts the one-step-back/one-step-forward principle. In essence, this architecture differs from the architecture of Figure 5-3 in that there is no need for a central repository of transparency data. Instead, the food operators are required to have “a system in place” that will enable them to identify the immediate supplier(s) and immediate customer(s) of their products. Since the regulations do not require electronic data sharing the food operators can be assumed to be of *type 1*.

5.3.3 An architecture following the EPCIS specification

An architecture that is gaining more attention in practice and research is one that is based on the EPCIS specification (EPCglobal 2014). The specification envisages a global distributed network of private systems deployed by businesses (Thiesse *et al.* 2009). It describes how transparency data, called EPCIS events, are captured and shared. As a result this architecture enables far greater transparency than the previous two but it also requires all food operators to deploy EPCIS-complaint private systems. Unlike the previous two architectures this architecture doesn’t require a central repository of transparency data, but, on the other hand, it depends on centralized discovery services that help locate where in the network the desired data are residing. The architecture involves only *type 4* and *type 5* food operators. Figure 5-5 depicts this architecture.

According to this architecture each food operator shall provide the necessary apps for its customers and business partners (shown as link *n*). As there are no 3p apps a food operator should, therefore, be capable of gathering and aggregating transparency data across the supply chain by querying each other’s private systems (*f*). As this architecture represents a global network (comparable to the internet) global services are provided by GS1 international and national organisations (which are depicts as a *3p* in the figure). GS1 services include discovery and master data synchronisation (GS1 2015).

Though a strict implementation of this architecture is hard to find in the meat sector some researchers have applied a hybrid of this and the centralized architecture. For instance, Shanahan *et. al.* (2009) proposed “enveloping” existing national cattle registration systems

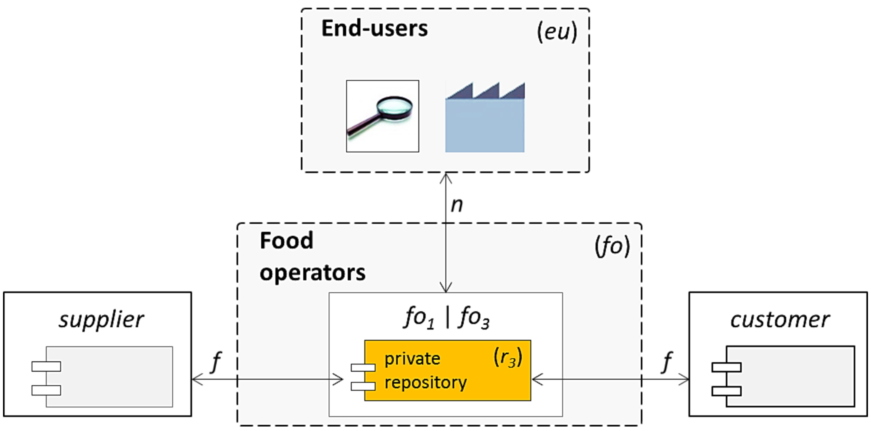


Figure 5-4. An architecture following the one-step-back/one-step-forward principle as an instance of the reference architecture

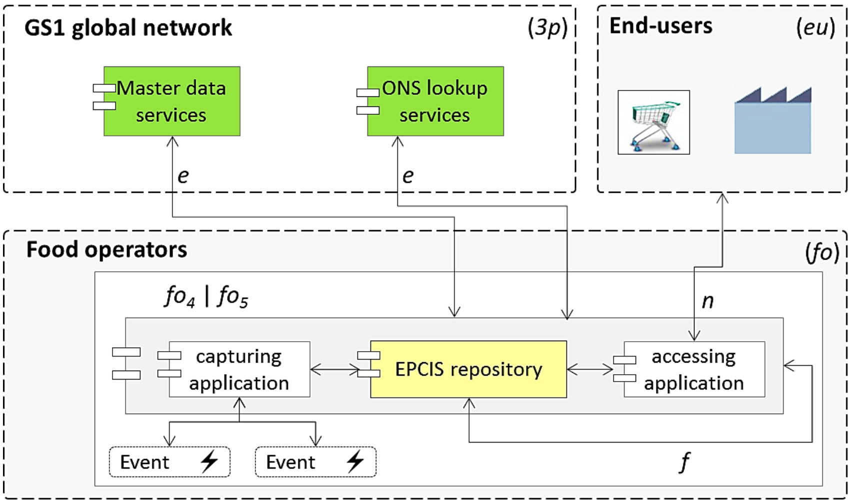


Figure 5-5. An architecture following the EPCIS specification as an instance of the reference architecture

with a software layer that makes them EPCIS-complaint. Other researchers have showed how some aspects of the EPCIS specification can be used in the meat sector. Feng *et. al.* (2013) described in detail the process steps of meat supply chains such as feeding, slaughter and segmentation. Thakur *et. al.* (2011) showed how such process steps in the food sector can be mapped to the Core Business Vocabulary (CBV, GS1 2014) used in the EPCIS specification.

5.3.4 Problem statement

It is possible to identify many more different alternative architectures depending on the level of transparency desired and the needs of the stakeholders involved. Usually, deriving a feasible concrete architecture that provides the desired features and level of transparency is difficult. Moreover, implementing an architecture is nontrivial and cumbersome. In light of this observation, we formulated the following two research questions:

1) *How to derive a feasible concrete architecture from the given reference architecture based on the level of transparency desired and the needs of the stakeholders involved?*

2) *How to realize the derived concrete architecture?*

5.4 Deriving a concrete architecture

In section 4.1 we introduce the systematic approach for deriving concrete architectures from the reference architecture. Subsequently in section 4.2 we illustrate the application of the approach by deriving a concrete architecture for CMTS.

5.4.1 Approach for deriving a concrete architecture

In Figure 5-6 we present a UML activity diagram representing the steps for deriving a concrete architecture from the reference architecture.

The first step is identifying the key stakeholders and their initial requirements based on the categories of stakeholders that have been defined in the reference architecture. The key stakeholders in meat supply chains are food operators, consumers, regulatory authorities, and third parties (Kassahun *et al.* 2014). In this step the specific stakeholders from each of the four groups have to be identified and their requirements gathered. Though requirements of one stakeholder may be influenced by requirements of another, the initial requirements can be inventoried independently.

In the second step, the requirements from the different perspectives are refined and organised into a consistent set of requirements. A key element of refining and organising requirements is determining the level of transparency all stakeholders will agree with. This step is done iteratively with the previous step because the requirements from different stakeholders may contradict and need reconciliation. For instance, the requirements of food operators may not be compatible with the requirements of consumers or with constraints imposed by the technology chosen by the 3pTSP.

In the third step the components that will be part of the concrete system are selected. The basic components are already defined in the reference architecture, but they will be elaborated to meet the requirements of the concrete system. Additionally, new components that have not been identified in the reference architecture but are specific to the concrete system might need to be introduced. The components to be selected depend on a number of factors, of which the most important are the architectural style (centralized, distributed or hybrid) and the transparency standards chosen. In all cases transparency requires repositories for capturing and storing transparency data and a set of services and applications for accessing the data.

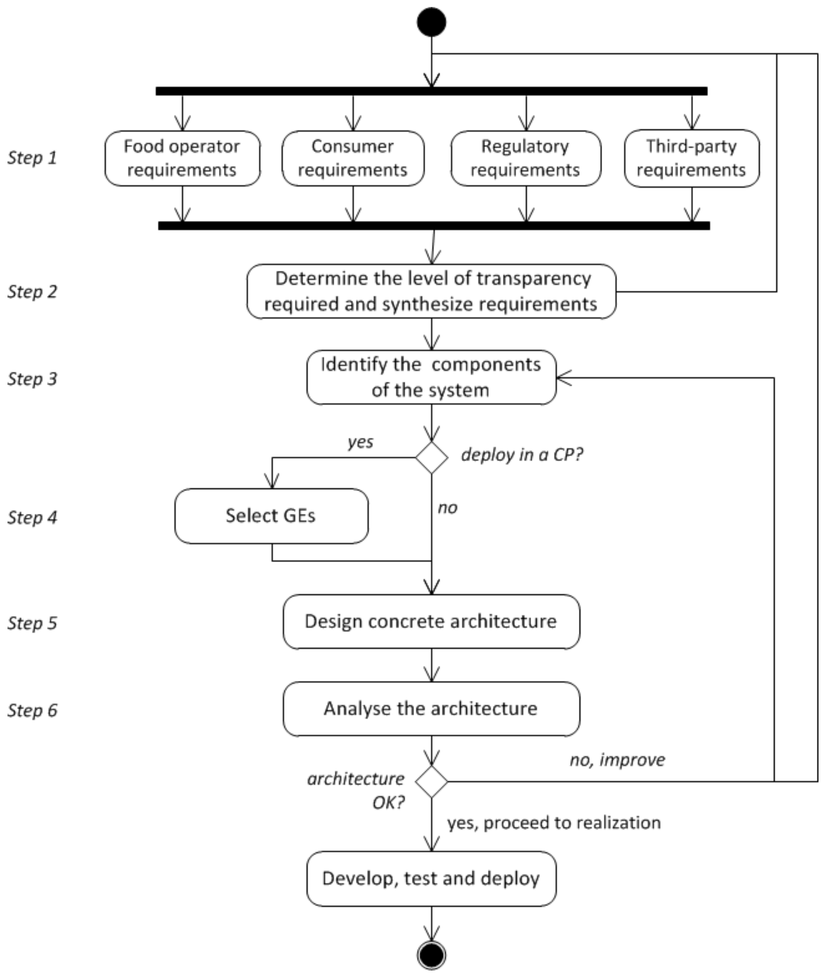


Figure 5-6. Activity diagram representing the process for deriving a concrete architecture from the reference architecture

In the fourth step the collaboration platform to be used is selected. Several collaboration platforms have been proposed in the literature. For example, FIspace is a collaboration platform for supporting business collaborations and includes ready-made components (Barmpounakis *et al.* 2015). To be useful the chosen platform should provide some or most of the generic components identified in the earlier step and it should facilitate the speedy implementation of the desired system.

In the fifth step the concrete architecture is design using the reference architecture and the set of requirements and components identified. At this stage the data models, the data and user interfaces of the components, and the interactions among the components are defined.

Finally in the sixth step, the architecture is analysed. If the resulting architecture is not adequate the architects might need to iterate back to the earlier steps. Once the architecture is finalized the realization of the system can proceed.

5.4.2 Deriving a concrete architecture for CMTS

The aim of CMTS is to achieve far greater transparency in the meat sector considering the current state of internal transparency systems at food operators. Meuwissen *et al.* (2003) noticed that the meat sector involves many small food operators that are unlikely to deploy state-of-the-art transparency systems within their premises as the costs largely outweigh the benefits. This fact will most likely not change in the near future. On the other hand, though global transparency standards (e.g. EPC-related standards) require the use of private transparency systems, the adoption of such standards is fundamental for realizing a truly chain-wide transparency. Considering these constraints the architecture shown in Figure 5-7 for CMTS has been derived.

As described in the activity diagram in Figure 5-6 the first two steps consist of identifying the stakeholders, their requirements and determining the level of transparency required. We considered a beef supply chain in Germany and, therefore, we considered the relevant European regulatory requirements. A range of organisations that provide transparency standards and procedures were consulted including GS1, Orgainvent, EHI, Global G.A.P., and Q+S. We chose to use EPC-related transparency standards. Requirements were gathered from GS1[2] Germany, food operators in Germany and consumers in Spain (through workshop sessions). We determined that the stakeholders wish to realize greater transparency (given a meat product at a retail shop they wish to trace it back to the farm and vice versa) but also that the majority of the food operators were unlikely to deploy a state-of-the-art transparency system in the near future and will remain *type 1* or *type 2* food operators. These steps were done in many iterations.

In the third step, we identified the desired components, which are a shared EPCIS repository (r_1) and three apps and one service (*sa*), to be provided by a 3pTSP (*3p*). Two of the three apps enable end-uses to query transparency data (*m*). The other app is used to capture transparency data manually by *type 1* food operators (*a*). The repository provides an API with which *type 2* food operators can transfer transparency data through an automated process (*b*). A *discovery service* is included for future support of *type 4* and *type 5* food operators –

[2] GS1 (http://www.gs1.org/) is a global non-profit organisation with over a million members. It develops and maintains standards for product identification and information sharing in supply chains.

allowing future extension for a hybrid architecture. The apps locate the EPCIS repositories using the discovery service (e_2). Data is queried using the query interfaces of the repositories (e_1).

In the fourth step, we considered the FIspace collaboration platform as a foundation for the transparency system. The FIspace platform aims to provide many crucial collaboration features, such as reusable apps, cloud hosting, security and identity management, data integration to link to external legacy systems (Barmpounakis *et al.* 2015). However, the platform turned out to be too complex. We decided to use only those features that will not affect the development and implementation of the system.

In the final steps, we designed and analysed the architecture of CMTS and issued an open call[3] to invite potential developers to realize it. CMTS was then implemented by a software developer with expertise in EPC-related standards. The detailed design and implementation – the demonstration of how this concrete architecture is instantiated – is presented in detail in the next section.

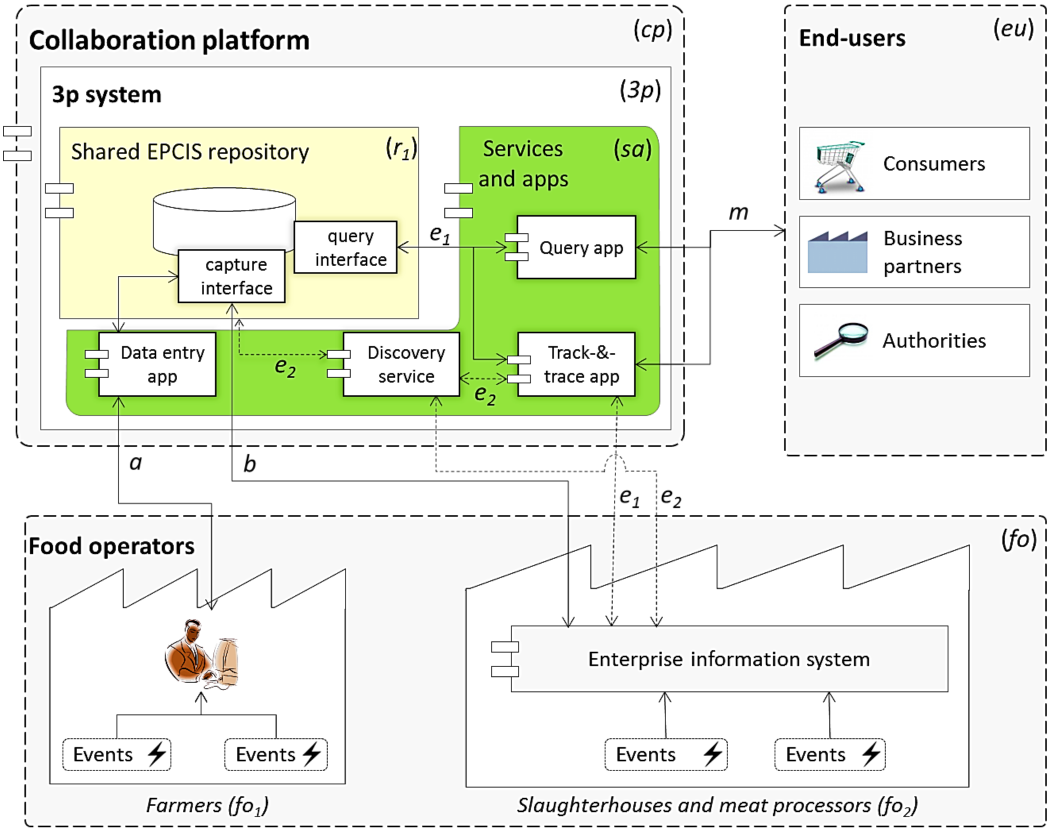


Figure 5-7. An architecture of CMTS derived from the reference architecture

5.5 Realization of the Transparency System

We instantiated the concrete application by realizing CMTS. CMTS is realized as a 3p system and is implemented as three web apps and one service built around a generic EPCIS system. The apps are a *data entry app* for manually capturing transparency data; a *query app* for

[3] http://fispace.eu/opencall.html

searching information about a specific product item locally at the shared EPCIS system and a *track & trace app* for searching information about a specific product item, its ingredients and where it is used as an ingredient across the supply chain. Besides a discovery service for discovering private EPCIS repositories is also realized. We used the Frequentz IRIS[4] transparency system as a generic EPCIS system.

Figure 5-8 and subsequent figures show screen shots of the 3p system. The four web links at the bottom of the figure provide the links to the three apps and the service. The farmer uses the *farmer* link to open a version of the data entry app with functionality that is limited to the data entry needs of farmers. The *aggregate* link points to the track & trace app. The *query* and *discover* links point to the query app and discovery service respectively.

We used the data entry app and automated data capture to populate the shared EPCIS repository with realistic transparency data. We use these data to describe CMTS in section 5.5.2 and section 5.5.3. In section 5.5.2 we describe the data capture functionality of the system, which includes the manual data entry and automated data capture. In section 5.5.3 we describe the query and track & trace query functionalities of the system. In section 5.5.4 we describe the discovery process along with the discovery service of the system[5].

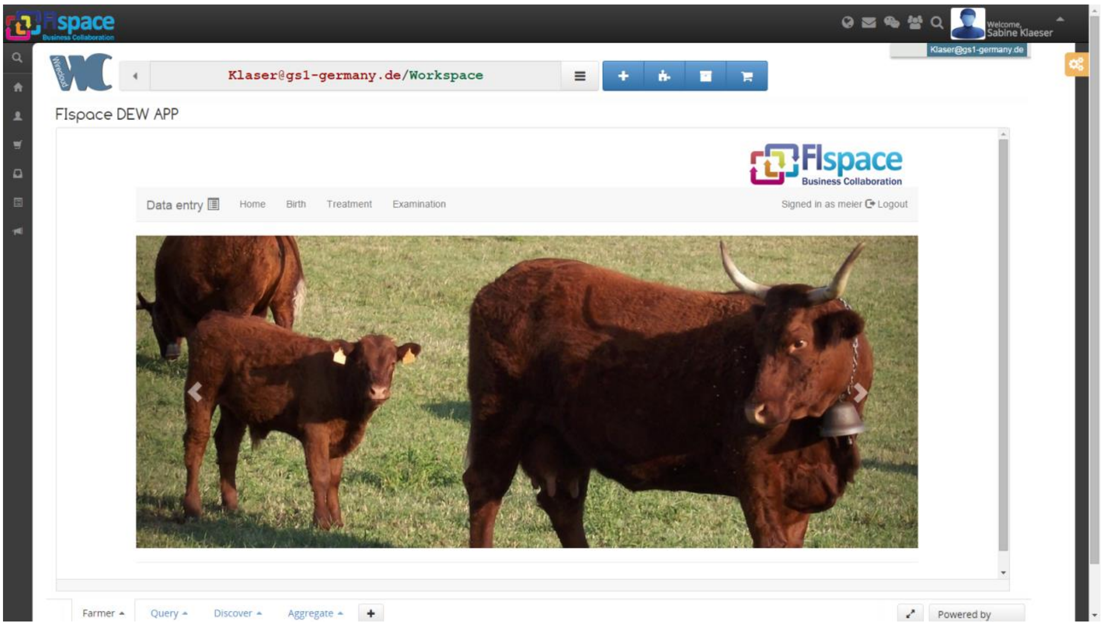


Figure 5-8. The main screen of CMTS[6]

5.5.1 The structure of transparency data

The EPCIS specification standardizes transparency data as EPCIS events and event data elements. The specification defines four types of events: *object*, *aggregation*[7], *transaction*, and *transformation* events. An EPCIS event is defined by data elements denoted as the four ‘dimensions’, which are the *identification* of the product(s) involved in the event, the *data and*

[4] http://frequentz.com/solutions/information-repository-intelligence-server/

[5] All data shown in the figures of subsequent sections are fictitious.

[6] The system is implemented by EECC, European EPC Competence Center GmbH in Cologne, Germany.

[7] *aggregation* event is not specifically related to the *aggregate* app.

time of the event, the *location* of the event and the *reason/context* of the event. These dimensions are conveniently called the *what*, *when*, *where* and *why* of EPCIS events (EPCglobal 2014, p32). Any data element that does not fit the four dimensions is captured as an *Instance/Lot Master-Data* (ILMD). ILMD is an extension mechanism that allows capturing specific and static attributes of individual product items and can optionally be included in an EPCIS event (EPCglobal 2014, p47).

In the following the use of the object and transformation events are demonstrated. The object event occurs when a product is created, as in the birth of an animal. The transformation event occurs when a product is processed as in splitting of carcass into sides and sections. The aggregation event that often occurs in relation to logistic operations and the transaction event that occurs when products are associated to business transactions are not dealt with in this study because the way these events get processed is not particularly different in meat supply chains from other cases.

Events are serialized as Simple Object Access Protocol (SOAP, Gudgin *et al.* 2007) messages as is described in the EPCIS *capture* and *query* interface specifications (EPCglobal 2014, pp 73 and 77). The SOAP messages are encoded as XML (eXtensible Markup Language) documents forming a SOAP XML message. The following sections present how the events are processed.

5.5.2 Data capture

Data capture initiates the creation of an EPCIS event. The object of the event (what) is identified by its EPC (EPCglobal 2014, p25) – a universally unique identification code of the product item. The EPC is constructed from a Global Trade Item Number (GTIN, GS1 2015, p144), which identifies a product (*e.g.* any half carcass of a particular type of animal at a particular food operator), and an additional serial or lot number, which identifies the particular item (*e.g.* the particular half carcass). The date and time (*datetime*, when) of an event is the datetime the event occurred. Usually this is the datetime at which the event is entered into the system by the capturing application. But, when data is entered manually, *i.e.* in case the event is captured after the fact, the datetime is the datetime the event occurred and not the datetime the event is manually recorded. The location (where) of an event is represented by a location EPC, which is mainly a Serialized Global Location Number (Serialized GLN or SGLN, GS1 2015, p91). The *why* of an event describes the business process step that is captured by the event. The process step is described by the CBV standard vocabulary (GS1 2014, pp 18-24). The CBV is generic and, therefore, does not define all business process steps encountered in the meat sector. Thus, CBV compatible (GS1 2014, p12) terms (such as medication and examination of animals) are defined and used.

5.5.2.1 Manual data entry

Manual data entry in the transparency system occurs when data is entered manually in a web form using a keyboard. It initiates the creation of an EPCIS object event. The web data entry form of CMTS provides many advantages over the paper-based data capture which is still widely being used today. The form is partially prefilled depending on previous activity and user preferences; the data entered is automatically validated to minimize data entry errors.

Figure 5-9 shows a data entry form of CMTS for registering the birth of a calf, which is one of the three manual data entry forms. (The other two forms are treatment and examination data entry forms.) To reduce data entry errors most of the form field values are validated either by: (1) providing only valid choices as drop-down selection list (*e.g.* a list of *Readpoints*), (2) checking the validity of the value entered (*e.g.* by checking the validity of a GTIN number), or (3) prefilling the field (e.g. the *Farm* is determined by the identity of the logged in user). Predefined values, such as drop-down selection lists, are fetched from a master-data registry. Visual feedback is provided for compulsory and incorrect form values. The form values are converted to an EPCIS event as described in section 5.5.2.3.

5.5.2.2 Automated data capture

Automated data capture occurs mainly in large slaughterhouses and meat processing plants by automated tag readers. This requires animals and meat products to be tagged with barcodes or RFID tags and these tags are read automatically. The data read by the tag readers are automatically converted to EPC (what). The date and time of reading is used as the datetime of the event (when). The location (where) and business context (why) are associated with the device that reads the tag. Any additional static information available in the RFID tag or elsewhere is captured as ILMD. Once these data are captured the subsequent processing of the data is similar to the processing of the web form data.

5.5.2.3 Web service interface for data capture

Data capture uses the EPCIS *capture* interface of the EPCIS system. The EPCIS specification specifies a single *capture* message. The capture message has one parameter which is a list of

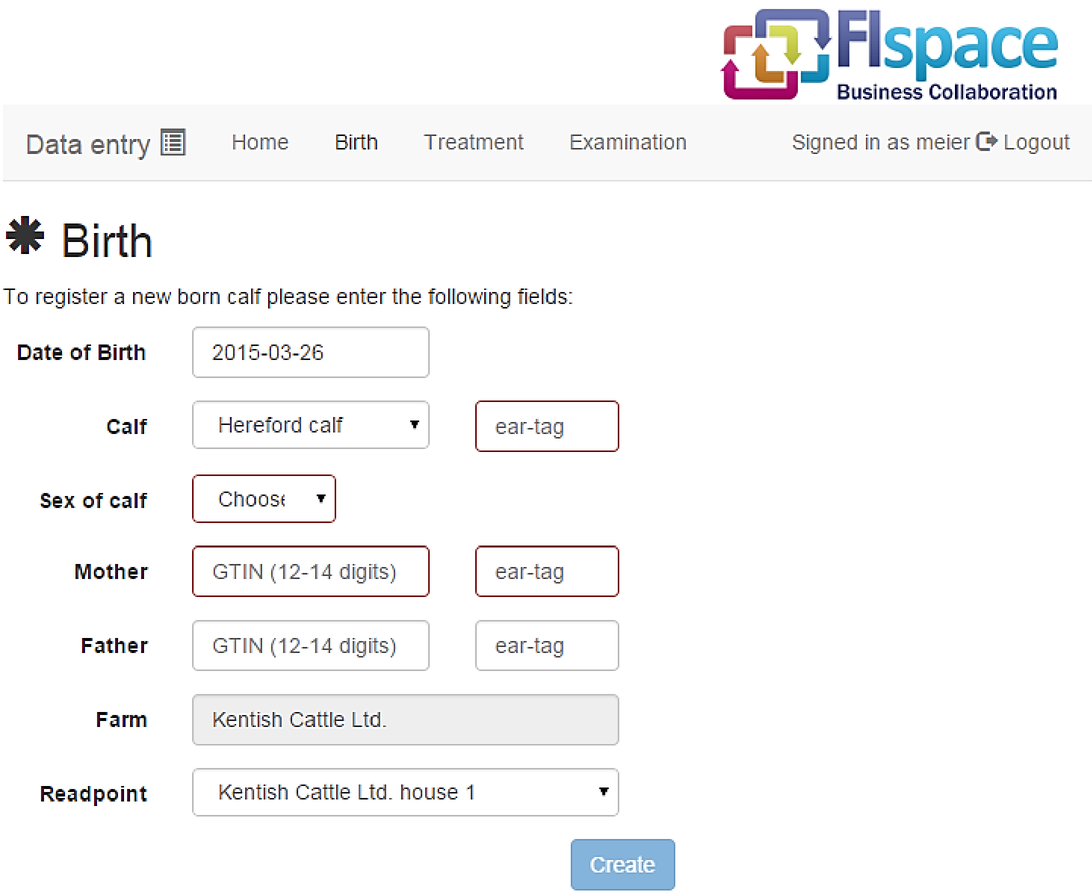


Figure 5-9. The data entry app user interface showing one of the manual data entry forms

EPCIS events serialized as a single SOAP XML (eXtensible Markup Language) document (EPCglobal 2014, p 73).

Figure 5-10 shows how the four dimensions and the ILMD of an EPCIS event representing the birth of a calf (a *birth* event) captured through the data entry app (the UI is shown in Figure 5-9) are serialized into a SOAP XML document. The document contains one EPCIS event inside the *EventList* XML tag. The *ObjectEvent* tag indicates that the birth event is captured as an *EPCIS object event*.

The *epcList* tag of an EPCIS object event contains a list of object EPCs (the *what* of the event) that are involved in the event. The figure shows that only one object – a newly born calf – was involved, therefore, there is only one *epc* tag inside the *epcList* tag. The *epc* tag contains a new Serialized GTIN (SGTIN, GS1 2015), urn:epc:id:sgtin:4023331.000714.ES01120999, that has been commissioned (created). The SGTIN is constructed from a GTIN, representing the breed of the animal at the specific farm, and a serial number, representing the specific calf. The GTIN is in turn derived from the company identification number (*company prefix*) of the farm and the *product number* the farm assigns to the breed. The breed of the animal read from the *Farm* and the *Calf* form fields are translated to a GTIN. The eartag assigned to the calf is used as a serial number. The translation of the values read from the form fields into GTIN and SGTIN is done using a master-data registry. The master-data registry maintains, among other things, the relationships between linguistic descriptions used in the form fields and the associated identification numbers. Thus, for instance, the *Kentish Cattle Ltd.* is converted to the company prefix 4023331and the breed *Hereford* is converted to the reference number 000714. The calf cannot yet be uniquely identified because all Herefords from Kentish Cattle will have the same GTIN. To uniquely identify the calf globally an SGTIN is generated from the GTIN and its *eartag* number. The *action* XML tag represents the action type (EPCglobal 2014, p 34) which can take either ADD, OBSERVE or DELETE as value. A newly commissioned EPC takes ADD as value for action.

The value of the *eventTime* and *eventTimeZoneOffset* tags represent the datetime (the *when*) of the event. The values were read from the *Date of Birth* form field.

The *bizStep* XML tag represents the *business step* (the activity) that triggered the event, which in this case is *commissioning*. *Commissioning* signifies the creation of a new object that is assigned a new EPC. The *disposition* XML tag represents what the *disposition* (the state) of the object is after the event occurred. All newly commissioned objects are in *active* disposition (*c.f. destroyed*, *inactive*, *expired*, *etc.*) indicating that an object "has just been introduced into the supply chain" (GS1 2014, p 26). These two tags together represent the business context (the *why*) of the event.

The *bizLocation* XML tag contains a list of locations as EPCs (the *where*) recorded under the *id* XML tags. In this case there is only one location and that is constructed from the *Farm* and *Read Point* form fields. The values of these form fields are converted to GLN and SGLN with the help of a master-data registry. The location of the farm is associated with a GLN and a specific read point in the farm is associated with an SGLN.

Additional information about the event is captured through the *extension* mechanism that is recorded inside the *extension* XML tag. The *extension* tag contains one *ilmd* element containing the values of the dam and sire of the calf.

The response XML signifies whether or not the request is successfully processed. However, the EPCIS standard does not specify a response SOAP XML document for the capture interface. A capture request is considered successful as long as HTTP “success” status code (code 200) is returned.

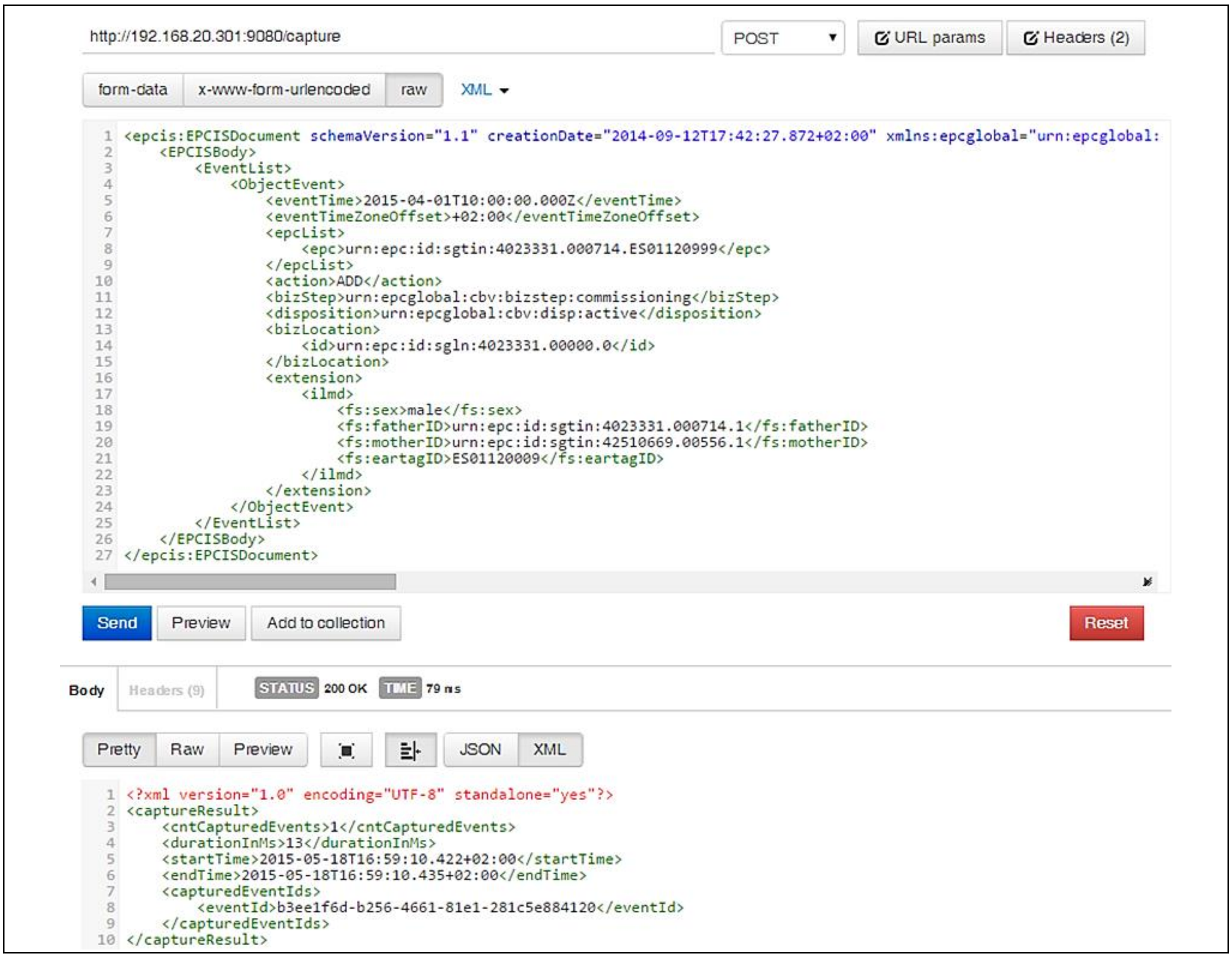


Figure 5-10. Capturing a birth event using the EPCIS capture interface of the shared EPCIS repository. The upper part of the image shows the SOAP request message to, and the lower part the SOAP response from, the EPCIS repository.

5.5.3 Data query

Data query is the opposite of data capture: given an EPC of an animal or a meat product item the query or the track & trace app fetches the events related to the EPC. Which events will be fetched depends on which of the two apps is used. The (ordinary) query app searches transparency data about the given product item only; on the other hand, the track & trace app makes a chain-wide query in search of transparency data about the given product item, all products from which the given product is made, and all products in which the product is processed. The chain-wide query is translated into a number of ordinary (item-specific) queries. Since a chain-wide query included ordinary queries only the track & trace app is described in the following sub-sections.

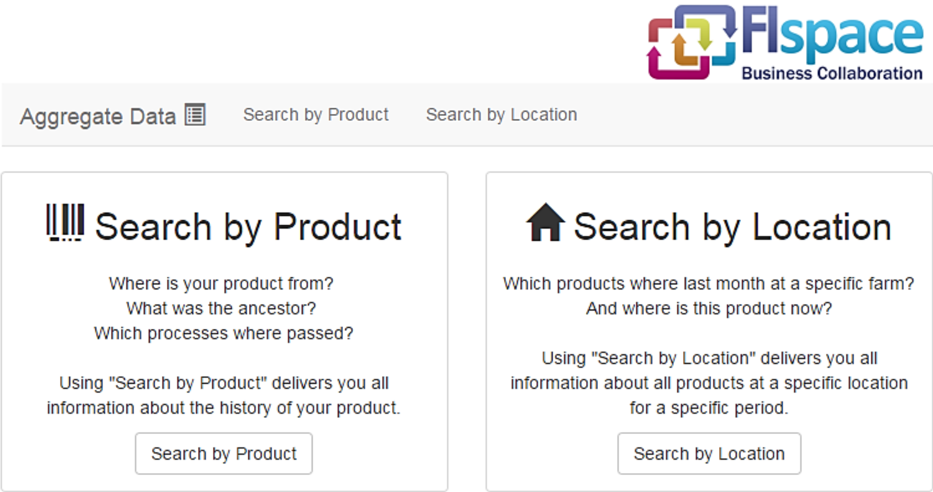


Figure 5-11. The two modes of the track & trace app: product- or location-based track & trace

5.5.3.1 The track & trace app

The track & trace app implements two ways of tracking & tracing: (1) search by product: track & trace a product given its EPC, and (2) search by location: track & trace all products that were at a given location and at the given time period (Figure 5-11).

An example of *search by product* is shown in Figure 5-12. Search by product takes an EPC – for instance, urn:epc:class:lgtin:426040435.0993.1 representing a meat product item at a meat processing factory – and results in: 1) the details of the product item such as the ID, serial number and description, 2) a list of EPCIS events representing the tracing of the product item, and 3) as a list of EPCIS events representing the tracking of the product item. The details constitute data on the four dimensions and any ILMD of the EPCIS event in which the EPC is commissioned. The tracing and tracking data is provided as a continuous list of EPCIS events. It is not always clear which events belong to *tracing* and which events to *tracking*. In most cases the events predating the commissioning of the EPC can be considered as tracing data, and the rest as tracking data (as the two separate lists shown in the figure). The figure shows the results of the track & trace of the EPC urn:epc:class:lgtin:426040435.0993.1 representing a particular chuck tender steak. The product item was traced back to 10 animals. In this particular case a batch processing was being simulated, therefore, the particular piece of meat could only be traced back to the 10 animals that were slaughtered and processed as a batch, but not to a particular animal. The particular steak was commissioned on September 9th and all EPCIS events up to the commissioning of the EPC are considered to constitute the tracing of the product item. All other EPCIS events in the list, which took place on the same or later date of the commissioning of the EPC are considered as the tracking of the product.

FIspace Business Collaboration

Aggregate Data | Search by Product | Search by Location

Search by Product

To get information about the history of your product, please enter the product ID:

Product ID

Search

Results

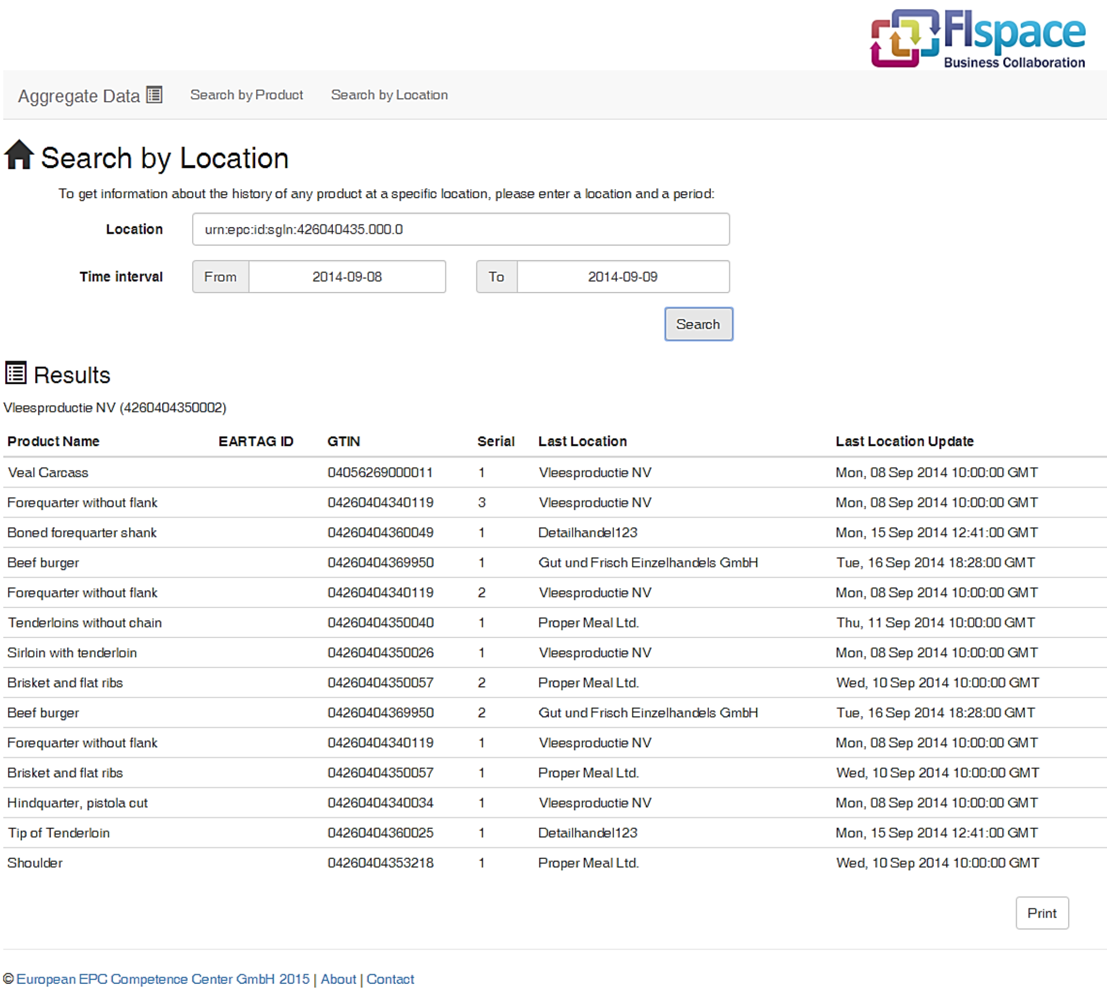
Product ID	urn:epc:class:lgtin:426040435.0993.1	**Expirationdate**	2014-09-23
GTIN	04260404359937		
Serial	1		
Description	Chuck tenderloin		
Owner	Vleesproductie NV		

Type	Date	Location	Items	
Birth	Fri, 01 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Sat, 02 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Sun, 03 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Mon, 04 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Tue, 05 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Wed, 06 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Thu, 07 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Fri, 08 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Sat, 09 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Birth	Sun, 10 Mar 2013 11:00:00 GMT	Kentish Cattle Ltd.	1	Details
Treatment	Sun, 07 Jul 2013 10:00:00 GMT	Kentish Cattle Ltd.	5	Details
Treatment	Mon, 08 Jul 2013 11:00:00 GMT	Kentish Cattle Ltd.	3	Details
Examination	Thu, 15 Aug 2013 10:00:00 GMT	Kentish Cattle Ltd.	10	Details
Production In	Wed, 03 Sep 2014 10:00:00 GMT	ABC Meat Ltd.	10	Details
Production Out	Wed, 03 Sep 2014 10:00:00 GMT	ABC Meat Ltd.	1	Details
Production In	Tue, 09 Sep 2014 10:00:00 GMT	Vleesproductie NV	1	Details
Production Out	Tue, 09 Sep 2014 10:00:00 GMT	Vleesproductie NV	1	Details
Shipping	Tue, 09 Sep 2014 22:15:00 GMT	Vleesproductie NV	1	Details
Receiving	Wed, 10 Sep 2014 04:34:00 GMT	Dover Sea and Land ...	1	Details
Shipping	Sun, 14 Sep 2014 22:27:00 GMT	Dover Sea and Land ...	1	Details
Receiving	Mon, 15 Sep 2014 12:41:00 GMT	Detailhandel123	1	Details

Figure 5-12. An example of track & trace data for a meat product item. The images show the result of searching for urn:epc:class:lgtin:426040435.0993.1 – and EPC that represents a particular chuck tender steak.

The data displayed in Figure 5-12 are fetched as follows. The datetime and location of the events (2nd and 3rd column) are read directly from the *eventTime* and *bizLocation* entries of the events. The type of the event (1st column) is read from master-data registry that links the EPC to its meat sector specific event type. For instance, the EPCIS object event is registered as a birth event. Items (4th column) is the number of items (animals or meat products) involved in the given event. For instance, the *examination* event in the figure indicated that the 10 animals were examined together. The details button (5th column) provides a link that displays the complete event data. Thus, the list can be read as follows: the product (a calf) was born on 4th of March at Kentish Cattle Ltd. (line 1); it was examined for BSE (the details shows the examination was a BSE test) on the 15 August, *etc*.

As example of search by location is shown in Figure 5-13. Search by location is important because in many cases of food contamination only the location and the time period of contamination is known, but not the particular product that caused the contamination. Location-based search requires the SGLN of the location (*e.g.* urn:epc:id:sgln:426040435.000.0) and the time period (*e.g.* between the 8th and 9th of September 2014) as search arguments. The search results in a list of product items, each of which is identified by its product name (1st column), GTIN (3rd column), the location where it



Product Name	EARTAG ID	GTIN	Serial	Last Location	Last Location Update
Veal Carcass		04056269000011	1	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Forequarter without flank		04260404340119	3	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Boned forequarter shank		04260404360049	1	Detailhandel123	Mon, 15 Sep 2014 12:41:00 GMT
Beef burger		04260404369950	1	Gut und Frisch Einzelhandels GmbH	Tue, 16 Sep 2014 18:28:00 GMT
Forequarter without flank		04260404340119	2	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Tenderloins without chain		04260404350040	1	Proper Meal Ltd.	Thu, 11 Sep 2014 10:00:00 GMT
Sirloin with tenderloin		04260404350026	1	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Brisket and flat ribs		04260404350057	2	Proper Meal Ltd.	Wed, 10 Sep 2014 10:00:00 GMT
Beef burger		04260404369950	2	Gut und Frisch Einzelhandels GmbH	Tue, 16 Sep 2014 18:28:00 GMT
Forequarter without flank		04260404340119	1	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Brisket and flat ribs		04260404350057	1	Proper Meal Ltd.	Wed, 10 Sep 2014 10:00:00 GMT
Hindquarter, pistola cut		04260404340034	1	Vleesproductie NV	Mon, 08 Sep 2014 10:00:00 GMT
Tip of Tenderloin		04260404360025	1	Detailhandel123	Mon, 15 Sep 2014 12:41:00 GMT
Shoulder		04260404353218	1	Proper Meal Ltd.	Wed, 10 Sep 2014 10:00:00 GMT

Figure 5-13. Tracking products that were at a given location and time period and all products that are made from them.

is last observed (5th column), the datetime of the last observation (6th column), and either the eartag (2nd column), if it is an animal, or the serial number (4th column), if it is a meat product. Search by location *traces* the current location of products that were at a given place and time and all products that are made from them.

5.5.3.2 Web service interface for query

The query and track & trace apps use the query interface of the shared EPCIS system and optionally other private EPCIS systems discovered by the discovery service. The EPCIS specification defines a query interface with seven query messages, of which the *poll* and *subscribe* messages are the most relevant (EPCglobal 2014, pp 77, 80). These two messages have a number of parameters of which the most relevant are *queryName* and *params*. The queryName parameter specifies the type of query of which there are two: *SimpleEventQuery* and *SimpleMasterDataQuery*. The params parameter contains a set of *param*/*value* pairs, where param is assigned predefined names, and value is assigned a value that is admitted for the given param. An example of such a predefined parameter name is MATCH_epc. This parameter name is used in our search queries and it means ‘match a given list of EPCs’. The allowed value for *value* is a list of EPCs that are to be searched.

The first part of Figure 5-14 shows the *poll* message as a SOAP XML document that is sent to the EPCIS system with SimpleEventQuery (EPCglobal 2014, p 90) as queryName and one param/value pair as params. The name of the param is MATCH_epc and its value is an EPC (which represents a Charolais calf). The second part of Figure 5-14 shows the response to the poll message. It contains two object events; the first an event registered when the calf was born (the birth event) and the second is an inspection event. In the second event the EPC was one of many other EPCs that took part in the inspection bizStep. The track & trace app builds the list of events such as that shown in Figure 5-13 by sending many poll messages to build the object history.

Request:

POST ▾ http://31.204.118.29:9080/services/EpcisService Go


```xml
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
    <Body>
        <Poll xmlns="urn:epcglobal:epcis-query:xsd:1">
            <queryName xmlns="">SimpleEventQuery</queryName>
            <params xmlns="">
                <!-- Optional -->
                <param>
                    <name>MATCH_epc</name>
                    <value>
                        <string>urn:epc:id:sgtin:42510669.00556.GB47520204</string>
                    </value>
                </param>
            </params>
        </Poll>
    </Body>
</Envelope>
```


Request Response

Response:


```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <epcisq:QueryResults xmlns:epcglobal="urn:epcglobal:xsd:1"
              xmlns:sbdh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
              xmlns:epcis="urn:epcglobal:epcis:xsd:1"
              xmlns:epcismd="urn:epcglobal:epcis-masterdata:xsd:1"
              xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1"
              xmlns:eecc="http://ns.eecc.info/epcis">
      <queryName>SimpleEventQuery</queryName>
      <resultsBody>
        <EventList>
          <ObjectEvent>
            <eventTime>2013-03-04T11:00:00.000Z</eventTime>
            <recordTime>2015-04-08T13:39:39.322Z</recordTime>
            <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
            <epcList>
              <epc>urn:epc:id:sgtin:42510669.00556.GB47520204</epc>
            </epcList>
            <action>ADD</action>
            <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
            <disposition>urn:epcglobal:cbv:disp:active</disposition>
            <bizLocation>
              <id>urn:epc:id:sgln:42510669.0000.0</id>
            </bizLocation>
            <extension>
              <ilmd>
                <fs:sex xmlns:fs="http://ns.fispace.eu/epcis">female</fs:sex>
                <fs:fatherID xmlns:fs="http://ns.fispace.eu/epcis">
                              urn:epc:id:sgtin:42510669.00044.304
                        </fs:fatherID>
                <fs:motherID xmlns:fs="http://ns.fispace.eu/epcis">
                              urn:epc:id:sgtin:42510669.00556.304
                        </fs:motherID>
                <fs:eartagID xmlns:fs="http://ns.fispace.eu/epcis">
                              GB47520204
                        </fs:eartagID>
              </ilmd>
            </extension>
            <eecc:eventId>a9612d1f-8112-4385-bb27-e9fb7bde9b9f</eecc:eventId>
          </ObjectEvent>
          <ObjectEvent>
            <eventTime>2013-08-15T11:00:00.000Z</eventTime>
            <recordTime>2015-04-08T13:39:39.568Z</recordTime>
            <eventTimeZoneOffset>+01:00</eventTimeZoneOffset>
            <epcList>
              <epc>urn:epc:id:sgtin:42510669.00556.GB47520201</epc>
```

```xml
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520202</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520203</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520204</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520205</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520206</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520207</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520208</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47520209</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522010</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522011</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522012</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522013</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522014</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522015</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522016</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522017</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522018</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522019</epc>
          <epc>urn:epc:id:sgtin:42510669.00556.GB47522020</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:inspecting</bizStep>
        <bizLocation>
          <id>urn:epc:id:sgln:42510669.0000.0</id>
        </bizLocation>
        <fs:examinationList xmlns:fs="http://ns.fispace.eu/epcis">
          <fs:examinationElement>
            <fs:examinationType>bse_test</fs:examinationType>
            <fs:comment>no significant findings</fs:comment>
          </fs:examinationElement>
        </fs:examinationList>
        <eecc:eventId>92e73257-7b29-4844-b122-517bb46e6a26</eecc:eventId>
      </ObjectEvent>
    </EventList>
   </resultsBody>
  </epcisq:QueryResults>
 </soap:Body>
</soap:Envelope>
```


Figure 5-14. Querying for transparency data using the EPCIS query interface. The upper part shows a SOAP request message sent to the shared EPCIS repository; the lower part shows the response.

5.5.4 Discovery

If, besides the shared EPCIS system, other private EPCIS-compliant systems are also part of the chain-wide transparency system, then the track & trace app has to discover the addresses of these EPCIS systems so that it can query them for relevant EPCIS events. Given an EPC the discovery service provides information about these EPCIS systems as WSDL (Web Service Description Language) documents (Figure 5-15).

The GS1 discovery services initiative aims to deliver a specification that allows "full supply chain discovery" (GS1 2015). When realized a GS1 discovery service will discover all EPCIS systems that have some data about a given EPC. Realizing such a global discovery has turned out to be very difficult and the Discovery Services specification is for years under development. GS1, however, provides the Object Name Server (ONS) specification that, given an EPC, allows discovering the origination EPCIS systems. The ONS service works following the same principle as the Domain Name System (DNS) (EPCglobal 2013) and requires a global infrastructure that is currently not easily accessible.

The discovery service works, therefore, not on a global but on a supply chain scale and, as a result, doesn't require a global infrastructure. The discovery service, like the shared and on-demand repositories, is provided by 3pTSPs. Since all private repositories have to be registered at the discovery service by the food operators, given an EPC the discovery service can determine identify the EPCIS repository by extracting the company prefix (GS1 2015) from the EPC.

Figure 5-15. The discovery service showing how the addresses of EPCIS repositories are discovered given an EPC

5.6 Conclusion

Designing and implementing a chain wide meat supply chain transparency system is not trivial. This paper aims to provide guidance in the design and implementation process. The contribution of this paper is threefold. First of all, we have presented and discussed a reference architecture for chain-wide transparency systems for meat supply chains. The reference architecture can be used to derive many different concrete architectures with respect to different stakeholder requirements. We have shown 4 different examples of concrete architectures that are derived from the reference architecture. Second, we have provided a systematic approach for deriving concrete architectures. The approach guides software architects in identifying the stakeholders, defining the desired transparency levels, identifying the components of the chain-wide transparency system to be realized and evaluating the design. We have used this process to derive a chain-wide transparency system for beef supply chains. Third, we have illustrated how to implement a system based on the derived concrete architecture and demonstrated the functionality of the system.

The reference architecture and the corresponding guidance process is general and could also be used in other fresh food supply chains where many small farmers and large food processors are involved and where mixing and transformation of intermediate products takes place.

ACKNOWLEDGEMENTS

The research leading to this paper received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) within the FI-PPP's project FIspace under grant agreement No. 604 123. The authors acknowledge the individuals and companies involved for their support and the European Community for their funding. We would particularly like to acknowledge Nicolas Becker of the European EPC Competence Center (EECC) in Neuss, Germany, for realizing the software system demonstrated in the paper.

6

APPLYING BITA* TO DERIVE A REFERENCE ARCHITECTURE FOR ENVIRONMENTAL MODELLING COLLABORATION SYSTEMS[1]

[1] To be submitted under the title *Deriving a Reference Architecture for Realizing a Collaboration System for Supporting Environmental Modelling Projects* and authored by Kassahun, A. and Tekinerdogan, B.

Abstract

Environmental decision and policy makers increasingly demand a multidisciplinary approach to address environmental issues in order to improve the quality and acceptability of their decisions and policies. A multidisciplinary modelling study almost always requires multi-organisational collaboration for at least two reasons. First, modelling organisations generally specialize only in one or few disciplines of the required multidisciplinary expertise. Second, many other organisations representing different types of stakeholders are involved. The success of the collaboration depends on the integration of diverse modelling processes and IT systems across the collaborating organisations. Realizing integration is difficult, which had become apparent in a series of harmonization projects that were conducted in Europe from 2002 to 2009 due to the lack of appropriate design abstractions and a systematic approach to harmonization. Recently, we have developed the BITA* framework for aligning business processes and IT across multiple organisations. In this paper we reframe the harmonization problems as alignment problems and apply the BITA* framework to address them and thereby evaluate the suitability of the framework to address similar alignment efforts. We illustrate the BITA* approach using a retrospective case study as a running example.

Keywords*:* Multidisciplinary Collaboration, Multidisciplinary Research, Business-IT Alignment, Environmental Modelling, Integration

6.1 INTRODUCTION

Environmental decision-makers nowadays demand multidisciplinary environmental models in order to improve the quality and acceptability of their decisions. This also follows from the demand of environmental policy-makers who are mandating a multidisciplinary approach to address environmental issues. Multidisciplinary environmental models help to inform decisions on environmental interventions as well as derive policies that will guide those decisions. In Europe, this was reflected in the declaration of the Water Framework Directive (WFD, EC 2003). The WFD mandated, among other things, participatory, multidisciplinary, river-basin level and model-based studies on managing the water resources of Europe.

The multidisciplinary approach calls for multi-organisational collaboration because organisations engaged in environmental modelling generally specialize only in one or few disciplines. For instance, a modelling study may involve simulating floods, analysing groundwater flows, studying the impacts of diverse physical interventions (such as building dikes or drainage systems) and evaluating socio-economic measures (such as pricing and quota). There is generally no single modelling organisation that specializes in all these disciplines.

The deliverables of an environmental modelling study can be grouped in to types of deliverables. Simple environmental modelling studies provide recommendations directly. Decision makers can then decide whether or not to implement them. Advanced environmental modelling studies, which are increasingly becoming common, deliver integrated modelling IT system which are usually referred to as Environmental Decision Support System (EDSS, Rizzoli and Young 1997). EDSSs enable the evaluation of various options and, therefore, enable the decision makers to make better decisions. Moreover, EDSSs facilitate the formulation of environmental policies. The aim of the WFD has been supporting the development of EDSSs and methodologies for applying them (Arnold *et al.* 2005, Mysiak *et al.* 2011).

An environmental modelling study is generally done in a context of a modelling project. The coordination of the project activities and the integration of diverse IT systems used in modelling have to occur within the limited life time of the modelling study. The success of multidisciplinary collaborative environmental modelling study requires a multidisciplinary methodology that guides the collaboration and modelling IT systems (often referred to as modelling tools) that can easily be integrated into an EDSS.

Defining multidisciplinary methodology and developing an EDSS is very challenging. To cope with this problem and support the objectives of the WFD a series of research projects were conducted in Europe from 2002 to 2009. The projects focussed on harmonization of the interfaces of modelling IT systems and the methodologies of environmental modelling (Arnold *et al.* 2005, Mysiak *et al.* 2011). Harmonization of interfaces aimed at standardizing the interfaces of diverse modelling IT systems so that creating an EDSS becomes manageable. The modelling IT systems involved include simulation and optimisation tools and software libraries. The modelling IT systems involved also include modelling support systems (such as project management and scientific workflow systems). Hereafter, we call the IT systems in general as *modelling IT systems*. Harmonization of methodologies aimed at developing multidisciplinary modelling guidelines and quality assurance procedures. The guidelines and procedures considered for supporting the WFD were mostly formulated as step-by-step process specifications. We thus call them hereafter as *modelling collaboration processes*.

To support the harmonization effort the European Commission (EC) launched a large research programme called Catchment Modelling (CatchMod, Arnold *et al.* 2005). The CatchMod programme consisted of ten projects and led to harmonization to a limited extent, often within few interrelated disciplines. To achieve broader multidisciplinary integration the programme was followed by a number of very large integration projects (refer to Mysiak *et al.* 2011, pp 66-69 for the list of projects).

We were members of two harmonization projects, which are the HarmoniQuA project and a follow-up AquaStress project. We have designed, developed and tested modelling collaboration process models and related modelling IT systems in the context of these projects. As active members of the CatchMod and follow-up programmes we have learned that the harmonization effort was difficult and time consuming largely due to the lack of appropriate design abstractions and a systematic approach for harmonization.

We have recently developed a design frameworks called BITA* aimed at aligning business processes and IT systems across multiple collaborating organisations. BITA* is partially based on our experiences in the two harmonization projects. The harmonization problems addressed in the harmonization projects are similar to alignment problems addressed using BITA*. BITA* can thus be helpful for aligning (a term which we hereafter use instead of harmonizing) modelling collaboration processes and modelling IT systems. BITA* provides modelling abstractions for addressing three types of alignment concerns, which are called Business Process to Business Process (BP2BP), Business Process to IT (BP2IT) and IT to IT (IT2IT) alignment concerns. The framework provides the modelling abstractions for addressing these alignment concerns and a systematic approach for applying the modelling abstractions (Kassahun and Tekinerdogan, submitted, chapter 3). In this paper we apply

BITA* retrospectively to the two projects mentioned before and thereby evaluate the suitability of the framework to addressing alignment issues in similar collaborative environmental modelling studies.

The remainder of this paper is organised as follows. In section 6.2 we present a case study that we use as a running example throughout the paper. In section 6.3 we briefly describe the BITA* framework. In section 6.4 we apply the BITA* framework to the case study retrospectively. In section 6.5 we discuss the results and finally we make concluding remarks in section 6.6.

6.2 A CASE STUDY AND PROBLEM DESCRIPTION

In this section we present a case study based on the HarmoniQuA and AquaStress projects. In the HarmoniQuA project we developed a comprehensive modelling collaboration process model (called the HarmoniQuA guideline) and a modelling IT system to support the guideline (Kassahun *et al.* 2004, Refsgaard *et al.* 2005, Scholten *et al.* 2007). In the AquaStress project we extended the HarmoniQuA approach to create an Integrated Solution Support (I3S, Kassahun *et al.* 2008, Assimacopoulos *et al.* 2009). Using the case study we formulate a problem statement. The problem statement concerns the suitability of BITA* for addressing alignment concerns in multidisciplinary modelling studies.

6.2.1 Case Study: Multi-organisational Collaboration in Environmental Modelling

Collaboration in a multidisciplinary environmental modelling study involves a number of organisations. Usually one of the collaborating organisations assumes the project leadership responsibility and coordinates all activities. The other organisations provide modellers, reviewers, and informants. No single modelling organisation specializes on all disciplines of environmental modelling, and as a consequence, diverse modelling organisations are involved. Informants are generally stakeholders on whose behalf the modelling study is being conducted. Likewise, the different stakeholders come from different organisations. A modelling study is generally conducted in a project, which is defined by a project plan that includes the definition of milestones and deliverables. Unlike a business process model, a project plan does not specify the flow of activities. Instead only activity dependencies are specified and the project plan is marked by milestones that describe the progress of the project. A conceptual model of collaboration in multidisciplinary environmental modelling studies is depicted in Figure 6-1.

Many multidisciplinary environmental modelling projects suffered setbacks in the past due disagreements on which modelling process models to follow and the lack of integration among the diverse modelling IT systems. The lack of explicit modelling collaboration process models is identified as a major cause of quality assurance problems in modelling (Refsgaard *et al.* 2005, Scholten *et al.* 2007). Likewise, the lack of integration among models made it difficult to make environmental decisions or derive environmental policies at river-basin level (Argent 2004, Moore and Tindall 2005).

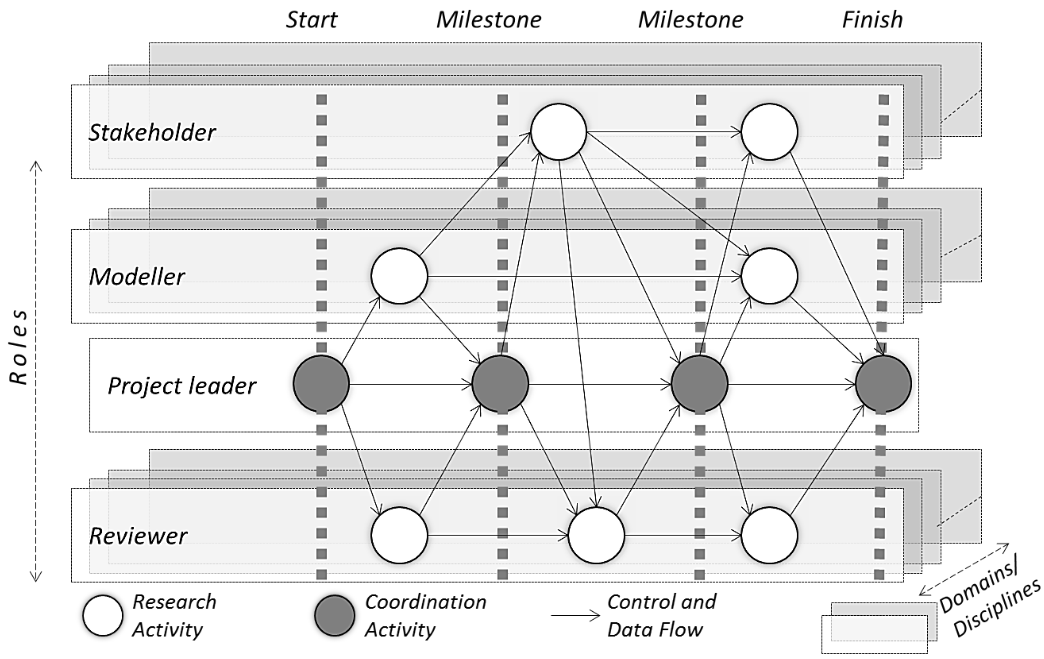


Figure 6-1: A conceptual model of collaboration in multidisciplinary environmental modelling studies

To illustrate the problem of alignment faced in the CatchMod projects we first use a process model shown in Figure 6-2 that represents a generic modelling collaboration process that is encountered in simple modelling studies. The modelling collaboration process consists of six generic activities: *plan*, *configure* (configure a model[2]), *run* (execute the configured model), *analyse* (analyse results of executing the configured model), *review* (review the model outputs) and *report* (report modelling results). The activities *plan*, *report* and *review* are common in many research and non-research projects and thus are self-explanatory. The three other constitute the conventional *configure-run-analyse* pattern of doing a modelling study, which we use as a running example in this paper.

The four vertical blocks (pools) of Figure 6-2 represent the organisations involved in modelling studies, namely a managing organisation (MA), an auditor (reviewer) organisation (AU), a modelling organisation (MO), and stakeholder organisation (SH). The last lane states that the activities are either executed by a modelling, stakeholder or both organisations. The three vertical bars displayed at the bottom-centre of the AU, MO, and MO|SH pools represent that generally multiple instances (OMG 2011) of these organisations are involved. There is generally only one managing (MA) organisation involved. It is important to realize here that the interactions among the modellers, among the stakeholders or among the auditors are not depicted. There is often little interaction among stakeholders or among auditors. The interaction among the modellers is, however, an important aspect of collaboration in modelling studies and has to be elaborated in most major modelling studies.

[2] It is important to realize here the following: (1) the term *model* here refers to environmental models and is unrelated to business process, IT and alignment models considered in the BITA* framework, and (2) *configuring a model* means making a generic model specific for the given problem, for instance, by setting parameter values.

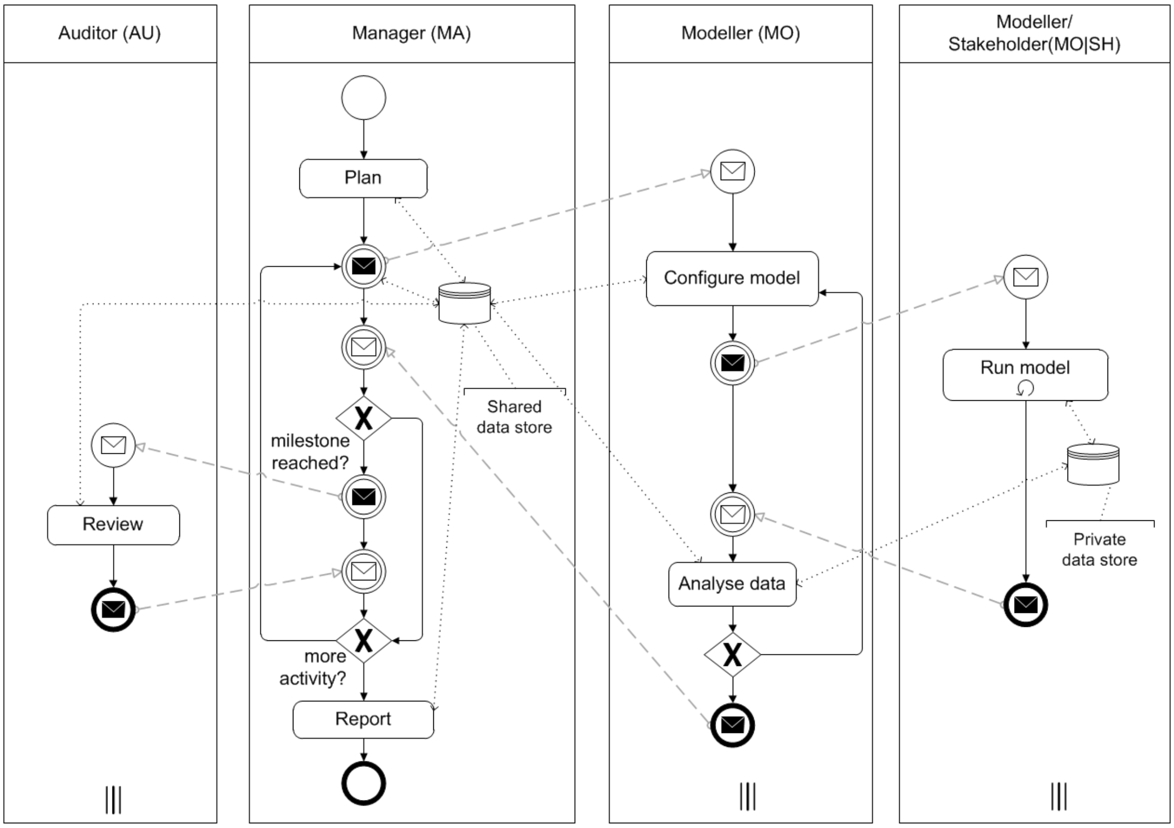


Figure 6-2: A generic modelling research collaboration process

6.2.2 Problem Statement

Disagreements on which modelling collaboration process model to follow and the lack of integration among modelling IT systems become obvious when we derive a more detailed version of the modelling collaboration process given in Figure 6-2. To derive a detailed version of the generic modelling collaboration process model we need a specific modelling problem scenario. For that purpose we use a simplified narrative that represents the pilot studies of the AquaStress project (see Assimacopoulos *et al.* 2009). The narrative is based on the water-stress mitigation process described by Kassahun *et al.* (2008):

> Imagine a certain region where both the availability of water and its quality (water stress) is of concern. Water is used by a variety of users, such as farmers, households and a manufacturing plant. Besides, a certain volume of water with a certain level of quality should be maintained for good ecological status. Regional authorities launch a multidisciplinary environmental modelling study involving knowledge management experts (KM-team), system dynamics modelling experts (SDM-team) and multi-criteria assessment experts (MCA-team). In addition local authorities and the manufacturing plant are also members of the modelling study. Imagine also, for the sake of brevity, that project management and review processes of the modelling study present no alignment issues. The KM-team generates knowledge models through a participatory process. The knowledge models represent the shared understanding about possible water-stress options, and the criteria with which they can be evaluated. The SDM-teams applies system dynamic modelling to investigate the various intervention options identified by the KM-team. The SDM-teams works with the manufacturing

plant, which is the largest polluter, to generate various water management options to reduce pollution. The MCA-team assesses the different views of households on the different water saving strategies based on a predefined and agreed upon set of criteria. The options and criteria for the MCA assessment are, partially, provided by the KM-team. The results of the SDM and MCA team are also used to improve the definition of the options and criteria; therefore, the three teams are largely dependent on each other's results. This narrative is illustrated as a BPMN models in Figure 6-3.

Figure 6-3 shows three modelling teams who are led by three different modelling organisations. (Note that Figure 6-3 shows only a small fraction of a modelling collaboration

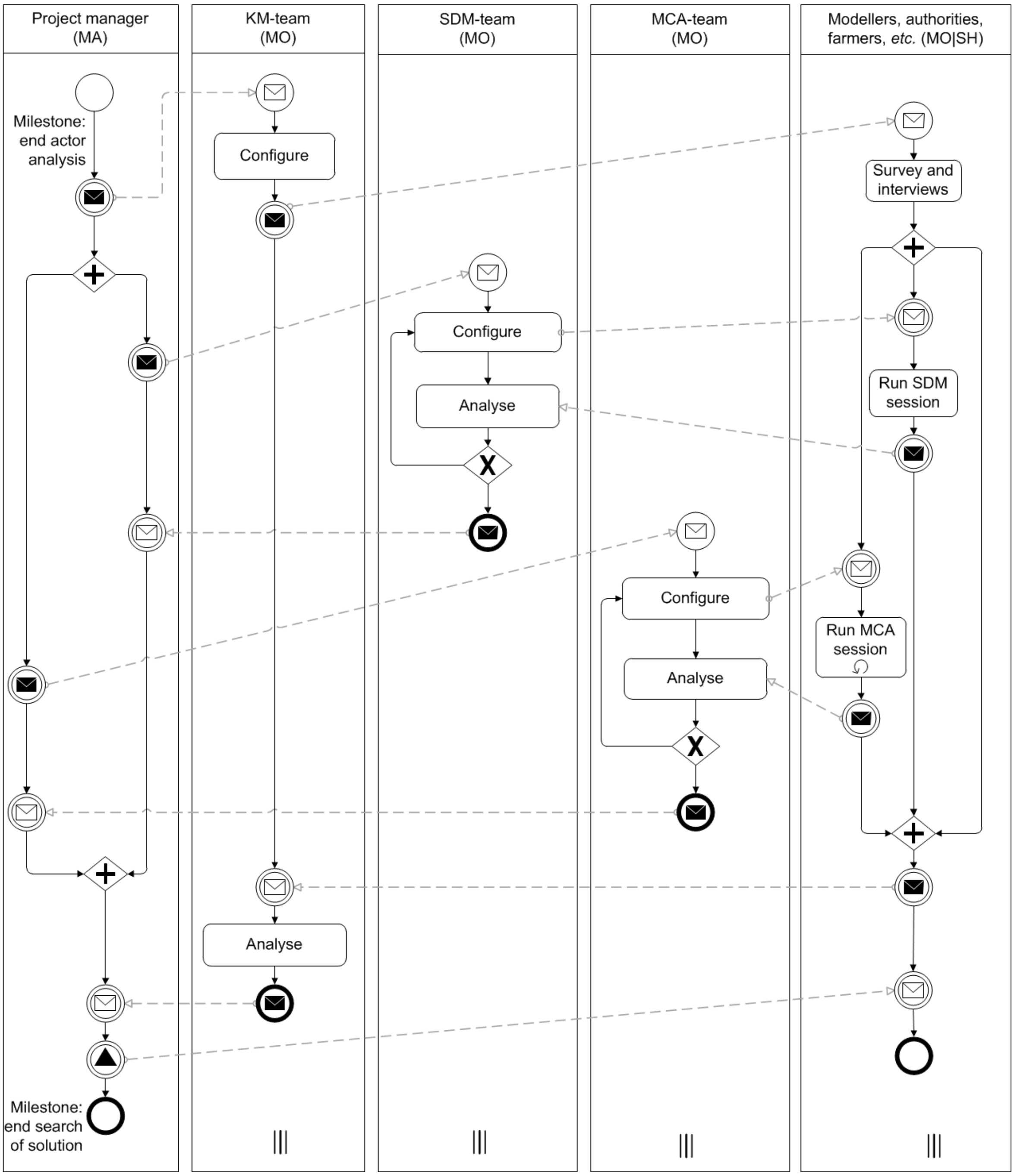


Figure 6-3. An example of a simplified water-stress mitiagation process inspired by the AquaStress pilot studies

process model that we encounter in modelling projects. A more complete model will be presented later in section 6.4.4.) In Figure 6-3 we derived the specific *configure*, *run* and *analyse* activities of the three modelling teams. When we compare Figure 6-3 with the actual activities of the pilot studies we realize misalignments. The KM-team does not “run” a model in any strict sense of model execution. In fact, the knowledge modelling (ontology modelling in the pilot studies) did not have a separate configuration and analysis phase. It involved instead many iterative deliberations with informants as well as other modelling teams—interactions that are not depicted in Figure 6-3. The teams had to collaborate on doing survey studies and modelling knowledge models (of options and criteria) but have different views on how to collaborate and achieve the required objectives. They also have different views on who should manage which data and who should provide which modelling IT systems.

In this paper we will illustrate that these problems are related to the BP2BP, BP2IT, and IT2IT alignment concerns identified in BITA*. We identify the specific alignment concerns and address them using the BITA* framework and thereby evaluate the suitability of the framework to addressing similar alignment issues in research collaboration.

6.3 Approach

6.3.1 The BITA Framework*

BITA* is a business-IT alignment framework for aligning business collaboration processes and IT systems across multiple collaborating organisations. The purpose of alignment is to help create a reference architecture that guides the integration of various business processes and IT systems adopted across a collaboration network. In BITA* a new reference architecture is derived from an existing generic reference architecture. The new reference architecture has to address the requirements of the specific sector or application area for which it is developed.

The framework provides three coherent design viewpoints, called BP2BP, BP2IT and IT2IT that target three types of stakeholders. The BP2BP alignment viewpoint is used by business analysts to align diverse business collaboration processes that are supported by the collaborating organisations collectively. The IT2IT alignment viewpoint is used by software architects to align the diverse IT systems that are distributed across the collaboration network. The BP2IT alignment viewpoint is used by an interdisciplinary team of business analysts and software architects to align the business collaboration processes with the underlying distributed IT system.

The core of the framework is a set of allocation and alignment models. Allocation and alignment models are used to generate allocation and alignment matrices. An allocation matrix represents a business collaboration process, a distributed IT system or the relationships between the two. An alignment matrix is a comparison of two corresponding allocation matrices. The framework provides a systematic approach that guides how to generate allocation and alignment matrices.

To help convert business collaboration models to allocation matrices the framework uses workflow patterns. Workflow patterns are recurring elements in business process models and the workflow systems that are used to execute business processes. In the past, more than a hundred workflow patterns have been identified, categorized as *control-flow*, *data-flow* and *resource-flow* patterns. A Control-Flow Pattern (CFP) defines a recurring pattern of activity sequencing in business processes. A Data Flow Pattern (DFP) defines a recurring pattern of data sharing in workflow systems. A Resource Flow Pattern (RFP) defines a recurring pattern of resource allocation in workflow systems (van der Aalst and ter Hofstede 2011). The framework used CFPs and DFPs only.

Each viewpoint consists of allocation and alignment models that address specific alignment concerns of the stakeholders of the viewpoint. The BP2BP allocation and alignment models represent BP2BP alignment concerns. The BP2IT allocation and alignment models represent BP2IT alignment concerns. The IT2IT allocation and alignment models represent IT2IT alignment concerns. Allocation matrices are filled in with binary (*true*/*false*) values. Alignment matrices are filled in with alignment attributes, which are *convergence*, *partial convergence*, *divergence*, *partial divergence*, and *absence*.

The alignment is not done pairwise (comparing allocation matrices of one organisation with another) but using a common generic reference architecture. Reference allocation matrices capture reference business collaboration models. However, each collaborating organisation adopts its version of business collaboration models (which is the cause for misalignment). The models adopted by the organisations are called concrete models. Thus, to each reference allocation matrix there can be as many concrete allocation matrices as there are collaborating organisations.

The concrete models are in many cases unavailable because many organisations do not maintain architectural documentation. This leads to two alternative approaches for generating alignment matrices.

1. The first, and an ideal, approach requires each organisation to produce concrete business collaboration models, from which concrete allocation matrices are derived. Alignment matrices are made by comparing a reference allocation matrix with the corresponding concrete allocation matrices. When an element in a reference allocation matrix is also present in all corresponding concrete allocation matrices; we indicate the element as *convergent*. When an element in a reference allocation matrix is absent in all corresponding concrete allocation matrices, we indicate the element as *absent*. If the element is present in some of the concrete allocation matrices, we indicate the element as *partially convergent*, which is equivalent to saying *partially absent*. When the reference allocation matrix does not contain an element that is universally present in all corresponding concrete allocation matrices, we indicate the element as *divergent*. But, if the element that is missing in the reference matrix is present only in some of the concrete allocation matrices, we indicate the element as *partially divergent*.
2. The second, and alternative, approach recognizes the fact that, in practice, concrete business collaboration models are unavailable. To circumvent this obstacle, alignment matrices are generated from formal reference models and informal concrete models (informal description of concrete scenarios). To generate an alignment matrix one of the following two approaches are possible. Either the reference allocations will be presented and explained to the representatives of each organisation, who will point out if the elements of the reference allocation table are convergent, divergent or absent with the tacit concrete model they have in mind. Or, the representatives will produce use case scenarios that describe the concrete models that those responsible for the alignment modelling can use to derive alignment matrices. The assignment of

convergence, *absence*, *partial convergence*, *divergence* and *partial divergence* attributes are made in the same fashion as in the first approach.

The second approach is followed in this paper. The alignments are based on analysing the use case description used in the HarmoniQuA and AquaStress projects (Groot *et al.* 2005, Kassahun *et al.* 2007, Kassahun *et al.* 2008, Assimacopoulos *et al.* 2009).

6.3.2 A Bootstrap Referecne Architecture

To apply BITA* a generic reference architecture is needed to start the alignment process. However, we were unable to find a suitable generic reference architecture for multidisciplinary modelling collaboration. Therefore, we must drive a reference architecture from relevant conceptual models.

For the given case study we define a generic reference architecture from a conceptual model for information systems provided by Piccoli's (2012), conceptual architecture for global software development provided by Yildiz *et al*. (2012), and from our experience and observarions in CatchMod and related projects.

According to Piccoli an information systrem of an organisation is a socio-technical system comprising of social and technical sub-systems. The social sub-system consists of structure and people; the technical sub-system consists of technology and processes. From multidisciplinary modelling study perspective, structure and people refers to modelling study teams (such as project management and modelling teams) and members of the modelling project (such as modellers or stakeholders); technology and processes refers to modelling IT systems and modelling collaboration processes. According Yildiz *et al*. global software development consists of a number of socio-technical systems as nodes interconnected through the internet. An important aspect of the interconnection is shared data, such as source code. In CatchMod and related projects shared data (Janssen *et al.* 2009), shared conceptualizations (ontologies) of modelling (Athanasiadis *et al.* 2009, Villa *et al.* 2009) and modelling processes (Scholten *et al.* 2007, Janssen *et al.* 2009) were considered as essential elements of collaboration in multidisciplinary modelling studies. Therefore, besides database (DB), we also add knowledge bases (KB) and process support (PS) to the conceptual architecture provided by Yildiz *et al*. to yield the generic reference architecture shown in Figure 6-4.

A reference architecture for collaboration often standardizes data and interfaces that enable the integration of IT systems. A thorough elaboration of data and interface standardizations used in multidisciplinary modelling studies is beyond the scope of this paper but is discussed in literature (Refsgaard and Henriksen 2004, Gregersen *et al.* 2007, Scholten *et al.* 2007, Athanasiadis *et al.* 2009, Villa *et al.* 2009). However, it is important at this stage to describe the relationships among the major data types and their relationships to DBs and KBs.

Figure 6-5 shows the major categories of data that we used in this paper. Note that we use the term data in the broadest interpretation of the term. The data used in modelling studies are classified into *process data* and *model data.* Process data are data about modelling activities, such as guidance, plans, project logs, and reports. Model data are data used in modelling activities, such as observed environmental data, model configurations, simulation runs, surveys and survey responses. Both process and model data can be classified into *qualitative* and *quantitative* data. Quantitative data are generally managed in relational databases and are structured. Examples of structured quantitative process data are process logs. Process logs consist of data about who has done what and when. Examples of structured quantitative model data are responses to closed-ended survey questions. Qualitative data can be classified into *structured* and *unstructured*. We use the term qualitative structured data to refer to what are commonly called semistructured data captured as ontologies, semantic networks or graphs. Examples of qualitative structured process data are guidelines. Guidelines can be structured into steps, activities, inputs, outputs, *etc.*, and all of these can be interlinked to one another. Examples of qualitative structured model data are specifications of criteria. In the AquaStress project, for instance, criteria are structured using the AquaStress criterion ontology. Qualitative unstructured data are all data that are not quantitative and not structured, such as images and documents. A project report is, for instance, a qualitative unstructured process data; an image (of a site, for instance) is an example of a qualitative unstructured model data.

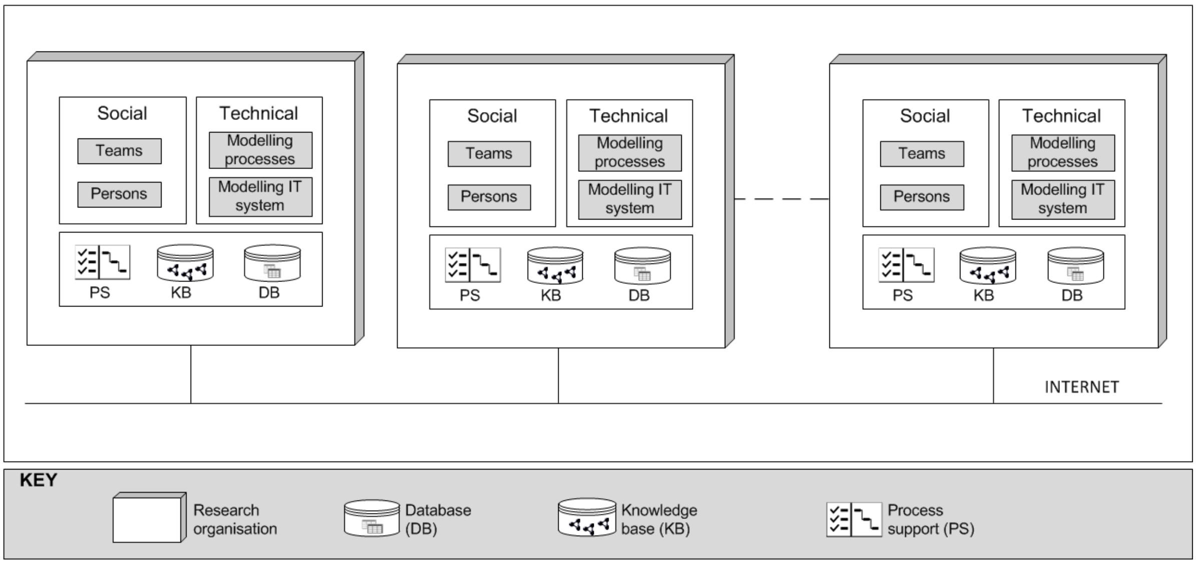


Figure 6-4: Reference architecture for research collaboration systems

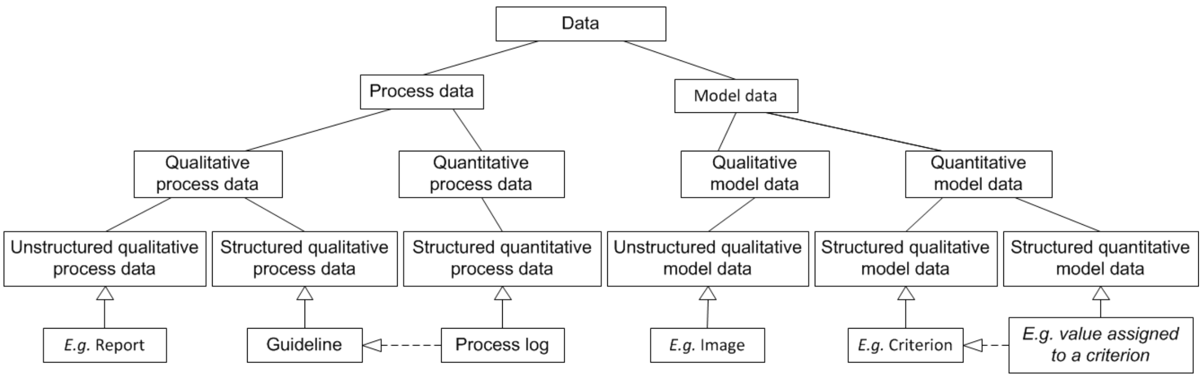


Figure 6-5: Feature diagram of the various data types used in environmental modelling study

Now that we have explained what the major data types are, we can describe how they are managed in DBs and KBs with the help of Figure 6-6. Structured data are, conventionally, managed using DBs; structured qualitative data are in recent years managed using KBs. Unstructured data are usually managed using document management systems but can also be managed using DBs. A DB in this paper refers to a collection of data that are structured following relational (Codd 1970) model. A KB refers to a collection of concepts and

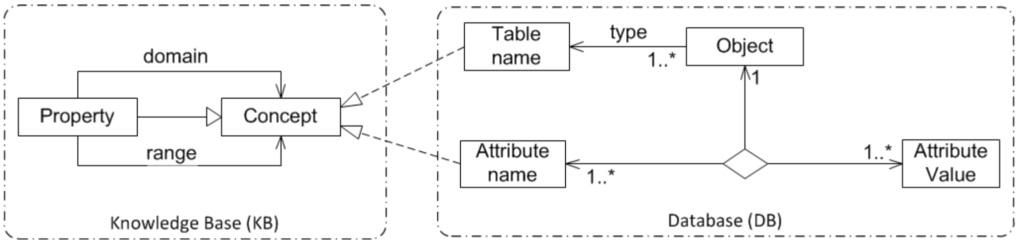


Figure 6-6: The relationship between the contents of DBs and KBs

terminologies interlinked with each other following ontological models (Noy and McGuinness 2001) or other formal semantic models.

A relational table contains rows that represent physical or conceptual objects. The objects (and thus the rows) are described by the attributes that constitutes the columns of the table. This implies that all objects of a table are of the same type and each object is described by the same attributes. The values associated with the attributes differ from one object to the other. The attributes are identified by the corresponding column names. The nature of the objects, in relational model, is defined by the name of the table, the name of the attributes and the values associated with the attributes. Table and attribute names are, however, not always sufficiently self-descriptive; and, therefore, the names are generally elaborated in detail elsewhere, often in documents or data dictionaries. But, ontologies are increasingly being used to provide a more explicit definition of concepts, including relational schemas (Martinez-Cruz *et al.* 2012, Spanos *et al.* 2012). In CatchMod-related projects ontologies were widely used to capture the explicit meaning of concepts which were only represented by names in relational databases, for instance, options and criteria. Ontologies are managed in KBs.

6.4 Applying BITA* to the Case Study

In this section we apply the three viewpoints of the BITA* framework to elaborate alignment concerns described in section 6.2 and address the concerns.

6.4.1 BP2BP View

6.4.1.1 BP2BP Allocation

In the BP2BP viewpoint of BITA*, activity and CFP allocations together capture business collaboration processes that are given as BPMN models.

Table 6-1 shows the reference activity allocations derived from the reference modelling collaboration process depicted in Figure 6-2. Basically, the activity allocation provides which organisation is responsible for which activities. This fact is represented in BPMN by placing an activity in a *pool*. A pool is used in this paper to represent an organisation. The activity allocation is simple because it is read directly from BPMN models. It is important to notice though, as in its BPMN counterpart, the column headings of Table 6-1 represent categories of organisations—and not the individual organisations as is often the case in BMPM models. The category MA represents the project management origination, of which there is usually only one. The category MO represents the modelling organisations, of which there can be many. The category AU represents the auditing organisations, which are generally few. The category SH represents other participating organisations, which provide data, informants or both.

Table 6-1: Reference activity allocations

Activities	*Organisations*			
	MA	*MO*	*AU*	*SH*
a1: plan	√			
a2: configure model		√		
a3: run model		√		√
a4: analyse data		√		
a5: review			√	
a6: report	√			

In Table 6-1 the control flows of Figure 6-2 are ignored, and thus, the table represents the modelling collaboration process only partially. To capture the control flows we use CFP allocations. CFPs can be arranged hierarchically; *i.e.* pattern representing the process model at higher level of abstraction contain patterns representing the details of the process model. As such, CFPs not only enable to capture control flows, they also enable to decompose complex processes flow into manageable and understandable sub-processes.

Because CFP allocations represent a hierarchy of CFPs, they are ideally represented as a diagram instead of a table. Figure 6-7 presents the CFP allocations that are derived from Figure 6-2. The same allocations are also shown in a tabular form in Table 6-2. We have identified in total seven CFP instances, which we number *p1* to *p7*. The second column (cfp_{id}) and third column (*name*) of Table 6-2 are the identification (*id*) and the name of the CFP as defined by van der Aalst and ter Hofstede (2011).

Pattern *p1* is a *sequence* (*cfp-1*) CFP. It represents the modelling collaboration process as four simple sequential steps: launch a modelling study (*p2*), plan the modelling study (*a1*), do all modelling activities (*p4*), report (*a6*), and end the modelling study (*p3*). The pattern *p2* is generally unqualified in BPMN models, but there are CFPs that allows us to qualify them. We model *p2* as a persistent trigger CFP (*cfp*-24), meaning that once launched the process must continue under normal circumstances. The alternative is transient trigger CFP (*cfp-23*), which means once started the process may be ignored for all sorts of reasons, for instance, a timeout. Likewise, the closure of a modelling study is modelled as explicit termination CFP (*cfp-43*)—the alternative being implicit termination (*cfp-13*). Implicit trigger and termination generally occur in short-lived processes.

The pattern *p4*, which represents all modelling activities, is modelled as a *structured loop* CFP (*cfp-21*). Like its containing CFP, *p4* represents the complex process within it (pattern *p5*) as a single element. Pattern *p4* contains *p5* and *a5*—the pattern *p5* is followed by the review modelling activities *a5*. If the reviewers are not satisfied yet another loop (iteration) is made, *i.e. p5* and *a5* are executed all over again; otherwise, the loop is broken and control passes to the containing CFP, which is *p1*.

The remaining three CFPs (*p5*, *p6* and *p7*) are modelled as composite patterns of multiple instance (*cfp-12*) and structured loop (*cfp-21*). They are multiple instance CFPs because they can be executed simultaneously by different modelling organisations. They are structured loop CFPs because all contain iterations.

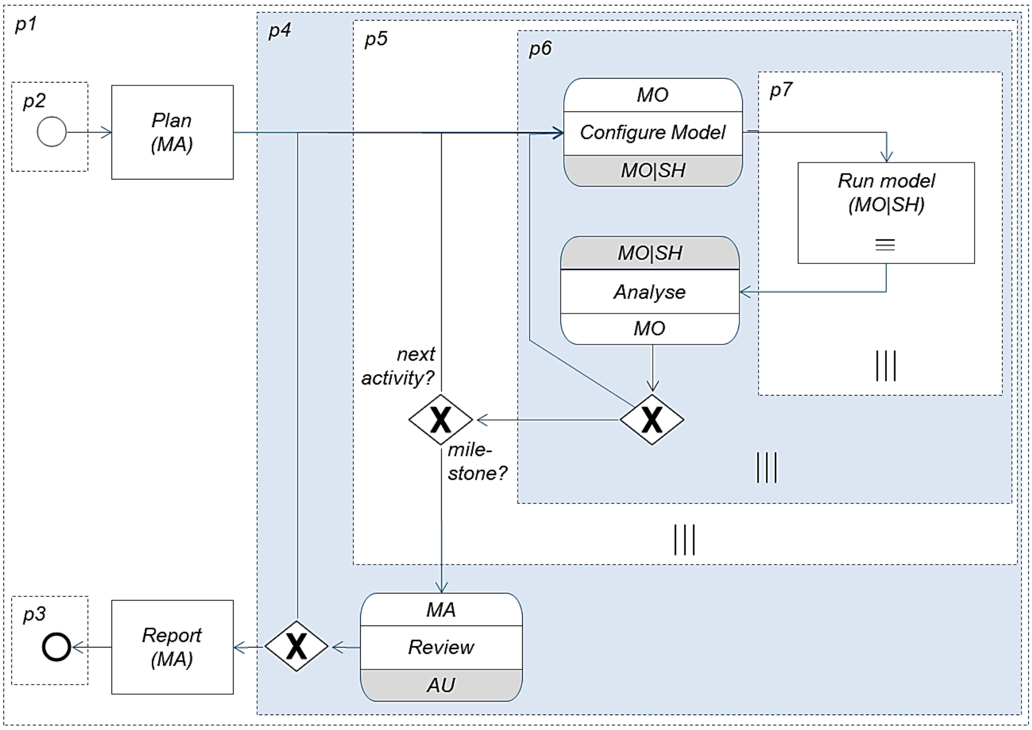


Figure 6-7: A CFP diagram for the reference collaborative query process model

Table 6-2: Reference CFP allocations

p_{id}	CFP			children					
	cfp_{id}	name	parent	a1	a2	a3	a4	a5	a6
p1	1	sequence		√					√
p2	24	persistent trigger	p1						
p3	43	explicit termination	p1						
p4	21	structured loop	p1					√	
p5	12+ 21	multiple instance + structured loop	p4						
p6	12+ 21	multiple instance + structured loop	p3		√		√		
p7	12+ 21	multiple instance + structured loop	p4			√			

6.4.1.2 BP2BP Alignment

Table 6-3 and Table 6-4 give activity and CFP alignments.

The activity alignments given in Table 6-3 are based on the more detailed reference modelling collaboration process model given in Figure 6-3, instead of the more generic counterpart given in Figure 6-2. As a result, more organisation types and activities are considered. The specific modelling organisations identified are: MO_{KM} representing the organisations of the KM-team; MO_{SDM} representing the organisations of the SDM-team; and MO_{MCA} representing the organisations of the MCA-team. The more specific activities of the three teams are indicated using additional number in the activity ID’s. The more detailed process model is considered in order to highlight alignment issues better.

The activity alignment table shows the following. The concrete activities related to SDM and MDA teams are convergent (+) with the reference activities. This means that the conventional *configure-run-analyse* pattern to modelling applies to SDM and MDA modelling.

But, the concrete activities related to the KM teams are either *divergent* (~) or *absent* (–).This means that the conventional *configure-run-analyse* pattern does not apply to the KM modelling. The activity *a3.2* (*deliberate KM*), which refers to deliberative and participatory process knowledge (ontology) modelling (Ferrand *et al.* 2007, Ribarova *et al.* 2011) is part of the concrete models but not the reference model, and is, therefore, *divergent* (~). Activities *a2.1* (*configure KM*) and *a4.1* (*analyse KM*) are predicted by the reference modelling collaboration proces, but in practice, no distinct configuration and analysis steps were specified by the KM teams of the given case study. Therefore, these two activities are *absent* (–) in the concrete models. We consider the activity *a3.1* (*elicit KM*), which refers to the knowledge elicitation process, generally using surveys and interviews, to be comparable to 'running a model', though a more close inspection shows substantial difference between running a conventional environmental model and eliciting ontologies through knowledge elicitation techniques. Particularly, ontology elicitation is a continuous process that is intertwined with the activities of SDM- and MCA- teams. Such a detailed level of alignment analysis is, however, not considered here.

Table 6-3: Activity alignments

Activities	*Organisations*					
	AU	*MA*	*MO_{ONT}*	*MO_{SDM}*	*MO_{MCA}*	*SH*
a1		+				
a2						
a2.1: configure KM			–			
a2.2:configure SDM				+		
a2.3:configure MCA					+	
a3						
a3.1:elicit KM			+			+
a3.2: deliberate KM				~	~	
a3.3:run SDM				+		+
a3.4:run MCA						+
a4						
a4.1:analyse KM			–			
a4.3:analyse SDM				+		
a4.4:analyse MCA					+	
a5: review	+					

Table 6-4 shows the CFPs alignments. The CFP alignments indicate that the concrete CFPs are generally in agreement with the reference CFPs, with the exception of pattern *p6*. This is due to the *absence* of activities *a2.1* and *a4.1*, and the *divergence* of activity *a3.2*.

It is worth to realize also that CFP alignment matrix has three dimensions. CFPs are related to other CFPs and activities. CFPs can be misaligned not only due to the misalignment caused by activities they contain, but also due to misalignments caused by their hierarchical arrangements (or misalignments of *parent* and *child* relationships). We make the distinction between the two by calling the former *content* misalignment and the latter *structure* misalignment. The pattern *p6* is not only misaligned content-wise; it is also misaligned in terms of structure. Namely, *p6* is *absent* in concrete CFP allocations.

Table 6-4: CFP alignments

CFP	Structural alignment	Content alignment					
		a1	a2	a3	a4	a5	a6
p1	+	+					+
p2	+						
p3	+						
p4	+						
p5	+					+	
p6	–		–		–		
p7	+			+			

6.4.2 BP2IT View

6.4.2.1 BP2IT Allocations

IT service and *I/O* allocations are defined in the BP2IT viewpoint of BITA* to represent the relationships among the elements of business collaboration process models, on one hand, and the elements of the business collaboration IT models, on the other. These allocations will be used here to represent the relationships among the elements of modelling collaboration process and IT models.

The modelling collaboration process considered is the one depicted in Figure 6-3. No explicit modelling collaboration IT model is available except the three IT systems (DB, KB, PS) mentioned in the bootstrap reference architecture. However, many diverse IT systems are involved in multidisciplinary environmental modelling studies. In the following we provide some of the modelling IT systems that were used in the given case study. We will also provide the SOA roles associated with the IT systems as part of the allocation modelling.

As in most research projects a questionnaire tool is used to gather data in environmental modelling studies. We refer to these tools as *Q-tool*. KM-teams use concept mapping and ontology development tools. For the given case study the CMap concept mapping tool (Cañas *et al.* 2004) and the Protégé ontology development tool (Gennari *et al.* 2003) were used. We refer to these tools collectively as *CM-tool* because they were mostly used for concept mapping. SDM-teams often use desktop simulation modelling tools. In the case study the Simile visual modelling environment (Muetzelfeldt and Massheder 2003) was used. We refer to these tools as *SDM-tool*. MCA-teams use diverse methods to elicit participants' assessment of various options and analyse the various traded-offs through Analytic Hierarchy Process (AHP, Saaty 2008). Eliciting participants' assessment and applying AHP directly is greatly simplified when an online multi-criterial assessment systems is used. For the case study we developed an MCA tool called AquaDT (Decision Tool for Water Management). We refer to such tools as *MCA-tool*. Modellers have to communicate the results of their study. The results are usually environmental decision support systems which are conveniently accessed online. For the given case study we developed an online system for displaying modelling results. We refer to such tools as query processing and presentation tool, or *QPP-tool*.

Table 6-5 shows the reference IT service allocations. The SOA role allocations provided in the table are based on our experience and insight. Above and in the generic reference architecture only the IT systems are identified. An IT system often provides a number of IT services. For the purpose of the IT allocation modelling, however, we considered the IT systems as monolithic IT services. It is important to realize, though, that the features of IT systems used in environmental modelling studies are increasingly being available as web services (Foster 2005). The IT systems described in the previous paragraph contain many features that could be offered as separate web services.

The reference IT service allocation shown in Table 6-5 is derived from the generic reference architecture given in Figure 6-4 and states the following. Each organisation has its own PS, KB and DB systems. The IT systems are used locally. It may be reasonable to assume that some of the organisations may share their DB and KB. But the reference architecture does not specify who the providers and who the clients are. Therefore, no SOA roles are associated to the three IT systems. The remaining five modelling IT systems described above are assigned SOA roles as follows. The *Q-tool*, *CM-tool*, *SDM-tool* and *MCA-tool* tools are *provided* by one or more modelling organisations. The *clients* of these tools are modelling and stakeholder organisations. The *QPP-tool* is generally *provided* by the managing organisation. *QPP-tool* is use to disseminate the results of modelling studies, therefore, all organisations are candidate *clients*.

Table 6-5: Reference IT service allocations

IT systems	*Providers*				*Clients*			
	AU	*MA*	*MO*	*SH*	*AU*	*MA*	*MO*	*SH*
PS								
KBs								
DBs								
Q-tool			√				√	√
CM-tool			√				√	√
SDM-tool			√				√	√
MCA-tool			√				√	√
QPP-tool		√			√	√	√	√

Next in the BP2IT allocation modelling is the allocation of inputs and outputs to activities. Once again we use our experience and insight to provide the reference data I/O allocations given in Table 6-6. For the sake of brevity we consider only model data, and not process data.

We identify five reference model data types that are central to KM, SDM and MCA modelling. The *question* (*q*) data object refers to structured qualitative data used in questionnaires and surveys. The *response* (*r*) data object refers to structured quantitative or qualitative data obtained from questionnaires and surveys. The *option* (*o*) data object refers to structured qualitative data used in SDM and MCA tools and identified largely by the KM-team. The *criterion* (*c*), also called *indicator*, refers to qualitative structured data used in MCA to evaluate options. The *value* (*v*) is used as collective name for quantitative data obtained through MCA and SDM modelling. Values are, for instance, data items associated to criteria in MCA assessment or data items assigned to feedback links in SDM modelling.

The reference I/O allocations are shown in Table 6-6. The allocation can be described as follows. There are no input and no output data to configuration of KM (*a2.1*) since the activity does not exist. Data inputs into configuration activities of SDM and MCA (*a2.2*, *a2.3*) are *options* and *criteria*; the data outputs are questions to be presented to informants. The data inputs into the *run* activities (*a3*) are the questions that are derived in the configuration phase. The data outputs of *a3* activities are generally *responses*. The data inputs of the deliberation activity (*a3.2*) are *options* and *criteria* derived in pervious SDM and MCA activities and the *responses* of the KB deliberation and yields improved *options* and *criteria* as output. There are no input and no output data to and from activity *a4.1* since the activity does not exist.

Data inputs to activity *a4* are *responses* and data outputs are *values*. The *values* are used to select relevant *options* and *criteria*.

Table 6-6: Reference I/O allocations

Activities	*Inputs*					*Outputs*				
	KB			*DB*		*KB*			*DB*	
	o	*c*	*q*	*r*	*v*	*o*	*c*	*q*	*r*	*v*
a2										
a2.1: configure KM										
a2.2:configure SDM	√	√						√		
a2.3:configure MCA	√	√						√		
a3										
a3.1:elicit KM			√						√	
a3.2: deliberate KM	√	√		√		√	√			
a3.3:run SDM			√						√	
a3.4:run MCA			√						√	
a4										
a4.1:analyse KM										
a4.3: analyse SDM				√		√	√			√
a4.4:analyse MCA				√		√	√			√

6.4.2.2 BP2IT Alignment

Table 6-7 gives the IT service alignments. We discuss the IT service alignments by dividing the table in to two sets of rows. The first three rows (IT systems) are the generic IT systems identified in the generic reference architecture given in Figure 6-4. The remaining rows represent the modelling IT systems described in the previous section (section 6.4.2.1). In the specific case study the managing organisation (MA) provided all the three generic IT systems for the modelling study participants, while most other participants kept their IT systems private. All actors (*clients*) used the PS system that the managing organisation *provided*. Modellers (*clients*) also used the KB and DB the managing organisation *provided* to share data. Some modelling organisations *provided* their DBs to other modelling organisations as a service (thus, some modelling organisations *clients* and others are *providers*). The alignment shows that IT service alignments involving the PS, KB, and DB are largely divergent (~). The other modelling IT services are, generally, in convergence (+).

Table 6-7: IT service alignments

IT services	*Providers*				*Brokers*	*Clients*			
	AU	*MA*	*MO*	*SH*	*MA*	*AU*	*MA*	*MO*	*SH*
PS		~			~	~	~	~	~
KBs		~						~	
DBs		~	#					~	
Q-tool			+					+	+
CM-tool			+					+	+
SDM-tool			+					+	+
MCA-tool			+					+	+
QPP-tool		+				+	+	+	+

Table 6-8 shows I/O alignments model data. The I/O alignments specify how the concrete data inputs and outputs to activities converge or diverge from the reference ones. The I/O allocations are straightforward, and generally convergent. Note that the activities that were *absent* or *divergent* (shaded in the table) in the activity alignment matrix are not considered for I/O alignment.

Table 6-8: I/O alignments

Activities	*Inputs*					*Outputs*				
	KB			*DB*		*KB*			*DB*	
	o	*c*	*q*	*r*	*v*	*o*	*c*	*q*	*r*	*v*
a2										
a2.1: configure KM										
a2.2:configure SDM	+	+						+		
a2.3:configure MCA	+	+						+		
a3										
a3.1:elicit KM			+						+	
a3.2: deliberate KM										
a3.3:run SDM			+						+	
a3.4:run MCA			+						+	
a4										
a4.1:analyse KM										
a4.3: analyse SDM				+		+	+			+
a4.4:analyse MCA				+		+	+			+

6.4.3 IT2IT View

6.4.3.1 IT2IT Allocations

IT system, *data object* and *DFP* allocations are defined in the IT2IT viewpoint of BITA* to model the distributed IT system. We have made no distinction between IT services and IT systems in the previous section, and as a result, no IT system allocation modelling is required here. In this section we consider *data object* and *DFP* allocations only.

Table 6-9 provides the data object allocations. As in the previous section we considered only model data. According to the bootstrap reference architecture each modelling organisation provides its own data management systems (DBs, and KBs); therefore, modelling data objects are allocated to the modelling organisations (MO).

Table 6-9: Reference data object allocations

Data objects	*Organisations*					
	AU	*MA*	MO_{KM}	MO_{SDM}	MO_{MCA}	*SH*
o:option			√	√	√	
c:criterion			√	√	√	
q:question			√	√	√	
r:response			√	√	√	
v:value				√	√	

DFP allocations describe the patterns of data sharing. There are four categories of DFPs, which are *visibility*, *interaction*, *transfer* and *routing*. In Table 6-10 we provide the reference DFP allocations partly based on the bootstrap reference architecture, partly based on our experience and insight. We discuss the DFP allocations using the four categories of DFPs.

Visibility: The bootstrap reference architecture is based on the idea that data is local and visible only with the scope of the instance of a modelling activity (*dfp-1*), and if the activity is iteratively executed, within the scope of the iteration (*dfp-4*). This is also consistent with the current practice in which each modelling organisation manages its own data and shares it only when it is needed (often by email).

Interaction: In the bootstrap reference architecture data interaction is either from activity to activity (*dfp-9*) or from activity to interactive (multiple instance) activity (*dfp-12, dfp-13*). Activity-to-activity data transfer means that the actors working on the current activity pass the resulting data directly to the actors working on the next activity. If the next activity is iterative the data should be stored or made continuously available until the iteration (multiple instance of the activity) finishes (*dfp-12*). If the data emerge from iterative activity (multiple instances of an activity) then the data must be gathered across the iteration (*dfp-13*).

Transfer: The two reference data transfer patterns are data transfer by by-value for inputs (*dfp-27*) and data transfer by by-value for outputs (*dfp-28*).

Routing: Data influences how the modelling collaboration processes "flows". Data values in SDM and MCA sessions influence how many iterations are made and thus affects the routing (dfp-40).

Table 6-10: Reference DFP allocations

DFP category	*DFP*	*q*	*r*	*o*	*c*	*v*
Visibility	*1*	√	√	√	√	√
	4	√	√	√	√	√
Interaction	*9*	√	√	√	√	√
	12	√	√	√	√	√
	13	√	√	√	√	√
Transfer	*27*	√	√	√	√	√
	28	√	√	√	√	√
Routing	*40*					√

6.4.3.2 IT2IT Alignment

In Table 6-11 and Table 6-12 show the data object and DFP alignments. The allocations of data objects were trivial and so are the alignments. In the given case study modelling organisations do generally manage their own data. The divergent (~) behaviour arises because the managing organisation provided shared KBs and DBs so that modelling organisations can share their by data centrally—instead of by various ad-hoc means, such as emails, as is often the case in modelling studies.

Table 6-11: Data object alignments

Data objects	*Organisations*					
	AU	*MA*	*MO_{KM}*	*MO_{SDM}*	*MO_{MCA}*	*SH*
o:option		~	+	+	+	
c:criterion		~	+	+	+	
q:question		~	+	+	+	
r:response		~	+	+	+	
v:value		~		+	+	

DFP alignment matrix is a three dimensional matrix (see Figure 3-11) consisting of *data object*, *DFP* and *organisation* dimensions. In the reference DFP allocations we did not consider the *organisation* dimension because the reference architecture suggests that all data are private and thus all organisations share data the same way. In the given case study the managing organisation provided shared data systems and shared data differently from modelling organisations. This makes the third dimension relevant. To incorporate the third dimension of the DFP alignment matrix we provided two sets of columns, one for MA and another for MO. The DFP alignment matrix is given in Table 6-12. We discuss the DFP alignments using the four categories of DFPs.

Visibility: The allocations *dfp-1* and *dfp-4* are in convergence since the reference allocations reflect the concrete reality in general. In the given case study shared KBs and DBs were introduced by MA. MA made modelling data visible to all members of the modelling team (*dfp-5*), or even people outside the project (external, *dfp-8*). The former is also enabled by the PS system (MoST) provided by MA and which was used as a project data sharing platform.

Interaction: The data interaction DFP allocations derived from the bootstrap reference architecture (*dfp-9, dfp-12* and *dfp-13*) are generally in *convergence* with the concrete data interaction DFP allocations. For the given case study, the introduction of shared DBs and KBs by MA leads to additional data interaction patterns, which are conveniently grouped into *read* concrete DFP allocations and *write* concrete DFP allocations. All data can be read (pulled) from an external source (DB) at once in the context of the entire project (*dfp-20*) or in the context individual activity (*dfp-16*). All data can be written (pushed) to an external source (DB) in the context of the entire project (*dfp-19*) or in the context individual activities (*dfp-15*). There are several other concrete interaction DFP allocations that are partially divergent (some modelling organisations practicing them while others not) but are here left out for brevity purposes.

Transfer: The two reference data transfer DFP allocations (*dfp-27, dfp-28*) are in convergence with their concrete counter parts. The two DFP allocations also apply to the shared systems provided by MA representing a divergent behaviour—not because the patters are new but because they apply to a new IT system introduced in the case study considered.

Routing: The only possible effect of data on control flow identified in the reference routing DFP allocations is *dfp-40*. However data influence routing in many other ways, particularly, if we consider how data can be used in PS systems to manage the control flow. However, the bootstrap reference architecture did not provide how data can be used in PS systems. In addition, the shared DBs and KBs provide additional possibilities of using data to manage the control flow. The concrete routing DFP allocations we describe next reveal insightful misalignments between the references and concrete allocation. Generally, data affect control flows (routing) either by just *existence* (when data come into existence something happens) or by its *value* (when the data have values that fall within some limits, something happens). The existence of a data object can affect either the *launching* (as precondition) or the *termination* (as postcondition) of an activity, or the *routing* of a gateway. The value of data object can, in a similar manner, affect the *launching* or the *termination* of an activity, or the *routing* of a gateway. Of all the various combinations we identified two DFP allocations in the given case study. Namely, for the given case, the existence of *question, response, option, and criterion* affected both the launching (*dfp-34*) and termination of activities (*dfp-36*).

Table 6-12: DFP alignments

DFP category	*DFP*	*MA*					*MO*				
		q	*r*	*o*	*c*	*v*	*q*	*r*	*o*	*c*	*v*
Visibility	*1*						+	+	+	+	+
	4						+	+	+	+	+
	5	~	~	~	~	~					
	8	~	~	~	~	~					
Interaction	*9*						+	+	+	+	+
	12						+	+	+	+	+
	13						+	+	+	+	+
	15	~	~	~	~	~					
	16	~	~	~	~	~					
	19	~	~	~	~	~					
	20	~	~	~	~	~					
Transfer	*27*	~	~	~	~	~	+	+	+	+	+
	28	~	~	~	~	~	+	+	+	+	+
Routing	*34*	~	~	~	~						
	36	~	~	~	~						
	40										–

6.4.4 Applying the alignment models

In HarmoniQuA and AquaStress projects modelling collaboration models were designed and implemented after a long process of harmonization. HarmoniQuA was conducted from January 2002 to December 2005 with a budget of €2.57 million. The project had 12 partners from 10 countries. The project delivered quality assurance guideline (called here modelling collaboration process model) and the corresponding IT tools (Kassahun *et al.* 2004, Scholten *et al.* 2004). AquaStress was conducted from February 2005 to January 2009 with a budget of €20 million. The project consisted of 35 partners, including universities, SMEs, NGOs and local water management authorities, from 17 countries. The project delivered integrated solution support system (distributed IT system) and associated multidisciplinary and participatory methodology for water stress mitigations (modelling collaboration process models) that were piloted in eight large test sites (AquaStress 2006). We present the modelling collaboration models developed in HarmoniQuA and AquaStress as reference architecture for multidisciplinary environmental modelling studies. We thereby reflect on how the BITA* approach would have facilitated the development of the reference architecture by comparing and contrasting the BITA* approach with the approach followed in HarmoniQuA and AquaStress.

6.4.4.1 Modelling Collaboration Process

The HarmoniQuA modelling guideline consists of over 350 activities (in contrast, Figure 6-3 identifies only a fraction of modelling activities) interconnected in complex control-flows that it will be difficult to depict it as a BPMN diagram. Instead, we grouped the activities hierarchically into 45 tasks, and the 45 tasks into 5 major steps. The tasks, the steps and the control flows among them are reproduced in Figure 6-8.

The HarmoniQuA modelling guideline harmonized the diverse disciplinary modelling guidelines to create a multidisciplinary modelling guideline. The resulting multidisciplinary guideline (shown in Figure 6-8) is also designed as cross-organisational process model. Therefore, we refer to it hereafter as the HarmoniQuA modelling collaboration process model.

The HarmoniQuA modelling collaboration process model is a result of harmonising a number of guidelines that are designed as imperative or declarative process models, including the Dutch the Good Modelling Practice (GMP, Van Waveren *et al.* 1999), Murrey-Darling groundwater flow modelling guideline in (Middlemis 2000), the Bay-Delta modelling protocol for water and environmental modelling in Californian (BDMF 2000), and many others (Refsgaard 2002).

Designing a collaboration process model not only requires identifying activities and control flows but it also requires assigning the activities to responsible organisations (in BPMN terms activities should be placed in pools). The HarmoniQuA modelling collaboration process identifies many types of organisations. The organisations can be grouped in various ways. The two prominent means are *roles* and *domain* of expertise. The different ways of grouping organisations are called customization tags. Likewise, as mentioned before, activities are grouped using various labels hierarchically. The customization tags, the activity labels, and various concepts used in the HarmoniQuA modelling collaboration process model are modelled as an ontology, which is depicted partially in Figure 6-9.

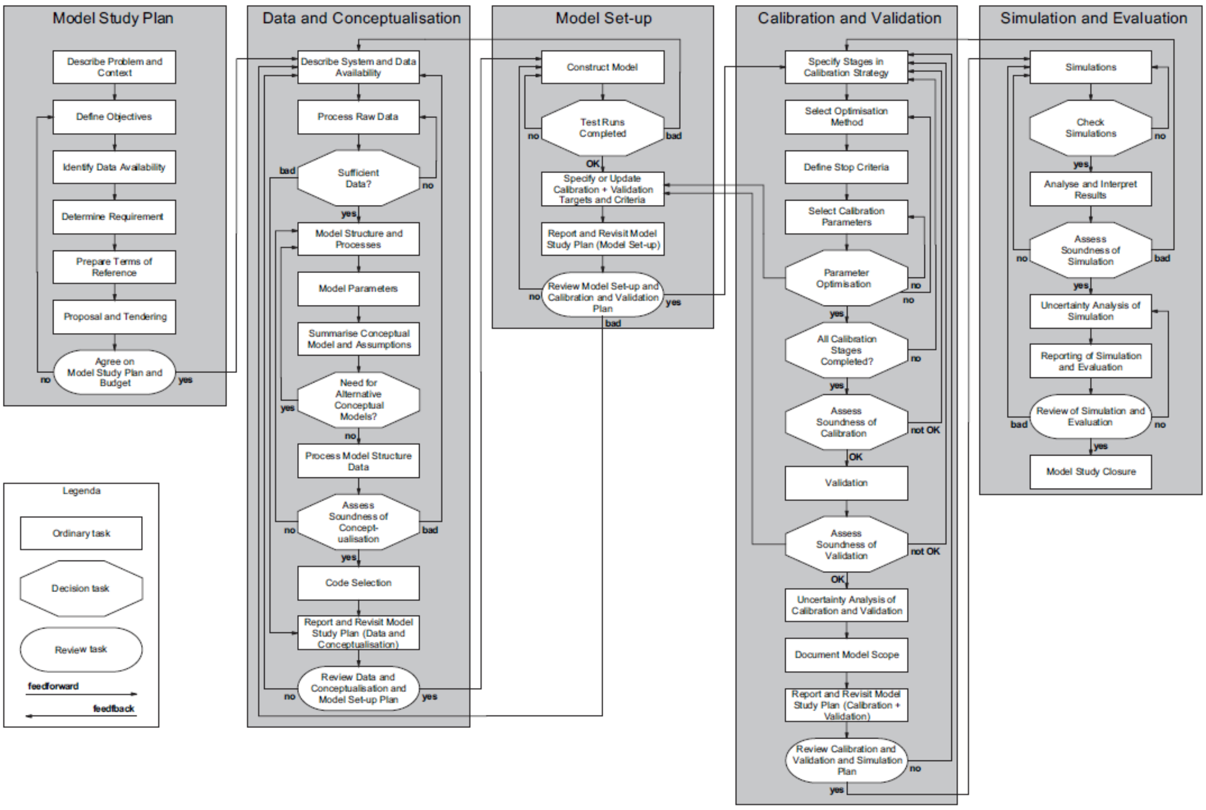


Figure 6-8. The HarmoniQuA modelling collaboration process (Scholten et al. 2007)

The HarmoniQuA modelling collaboration process model was developed by experts in envirmental modelling studies who come from the 12 organisations (and 10 countries) were divided into a large group of *modelling experts* and a small group (a *task force*) of modelling and IT experts. The modelling experts provided the detials of the modelling process model. The *task force* (of 4 environmental modelling experts and 2 knowledge engineering experts) designed the structure (ontology) the modelling process. The steps followed were as follows (Kassahun *et al.* 2007):

1. Modelling experts collected existing relevant modelling guidelines.
2. The task force analysed the existing guidelines and designed a guideline acquisition template.
3. Modelling experts used template to fill in the details of the multidisciplinary guideline.
4. The template and the guideline were evaluated and the steps 2, 3, 4 were repeated. The final guideline was established in 3 rounds of guideline development.

During the HarmoniQuA project the alignment models BITA* were not available. If they were available, the relevant viewpoint of BITA* will be BP2BP viewpoint, which is used in section 6.4.1.1. We believe that the CFP allocations (such as those given in Figure 6-7 and Table 6-2) and the corresponding alignments (such as that given in Table 6-4) would have revealed misalignments more quickly, if they were used during the HarmoniQuA project. Moreover, we believe that the relational models and the associated SQL queries for modelling allocation and alignment matrices given in Table 3-4 and Table 3-5 would have facilitated the development of the guideline acquisition template and the processing of the filled in templates.

The HarmoniQuA activities were structured using tasks and steps. The choices for tasks and

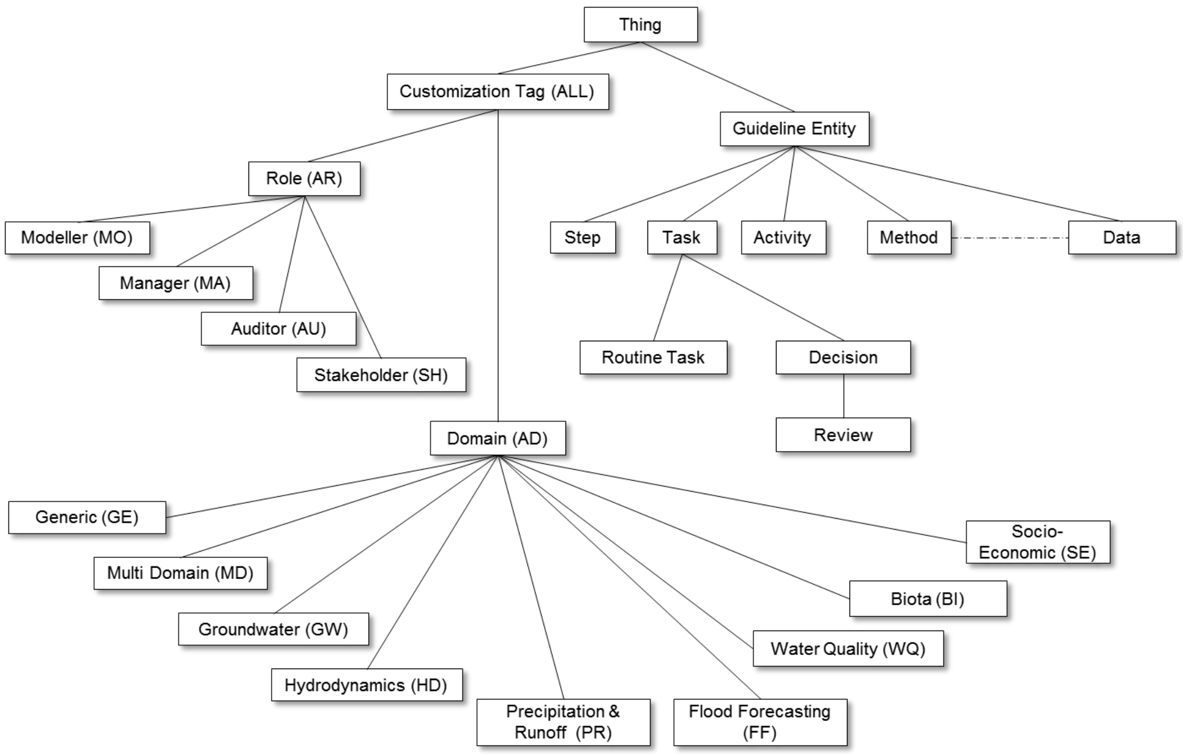


Figure 6-9. A feature diagram showing part of the HarmoniQuA guideline ontology (adapted from Kassahun et al. 2004).

steps were made largely based on the intuitions of the modelling experts. The use of more or less levels of abstractions and a different grouping of activities and task is also possible. In fact, much effort was put in agreeing on the level of abstractions and grouping of activities. We believe that the use of CFPs, and CFP modelling (such has those given in Table 6-2 and Table 6-4) might have made the discussions more objective and might have resulted in simplified and better control flow model.

6.4.4.2 Generic Modelling IT Systems

The bootstrap reference architecture (Figure 6-4) suggests each organisation might own PS (process support), DB (database), and KB (knowledge base) IT systems.

In HarmoniQuA a process support IT systems called MoST (Modelling Support Tool) was developed (Kassahun *et al.* 2004, Scholten *et al.* 2007). Figure 6-10 reproduces a screen shot of desktop version of MoST. MoST is a client server process support system that was partly following the workflow reference model (Hollingsworth and Hampshire 1993). As a workflow system it has a workflow engine which is the server of MoST and a worklist handler which is the desktop version of MoST. Unlike in other workflow systems, the process model used in MoST is fixed, and that is the HarmoniQuA guideline. The guideline is made specific for a given modelling study by filtering the guideline (which is very extensive) and removing the elements that are not relevant for the given modelling study (for a detailed explanation refere to Kassahun *et al.* 2004, Scholten *et al.* 2007). The MoST server includes a project log repository, which can be viewed as a DB, though it is not a relational DB. The project log represents *process data*. No *model data* DB is provided in HarmoniQuA.

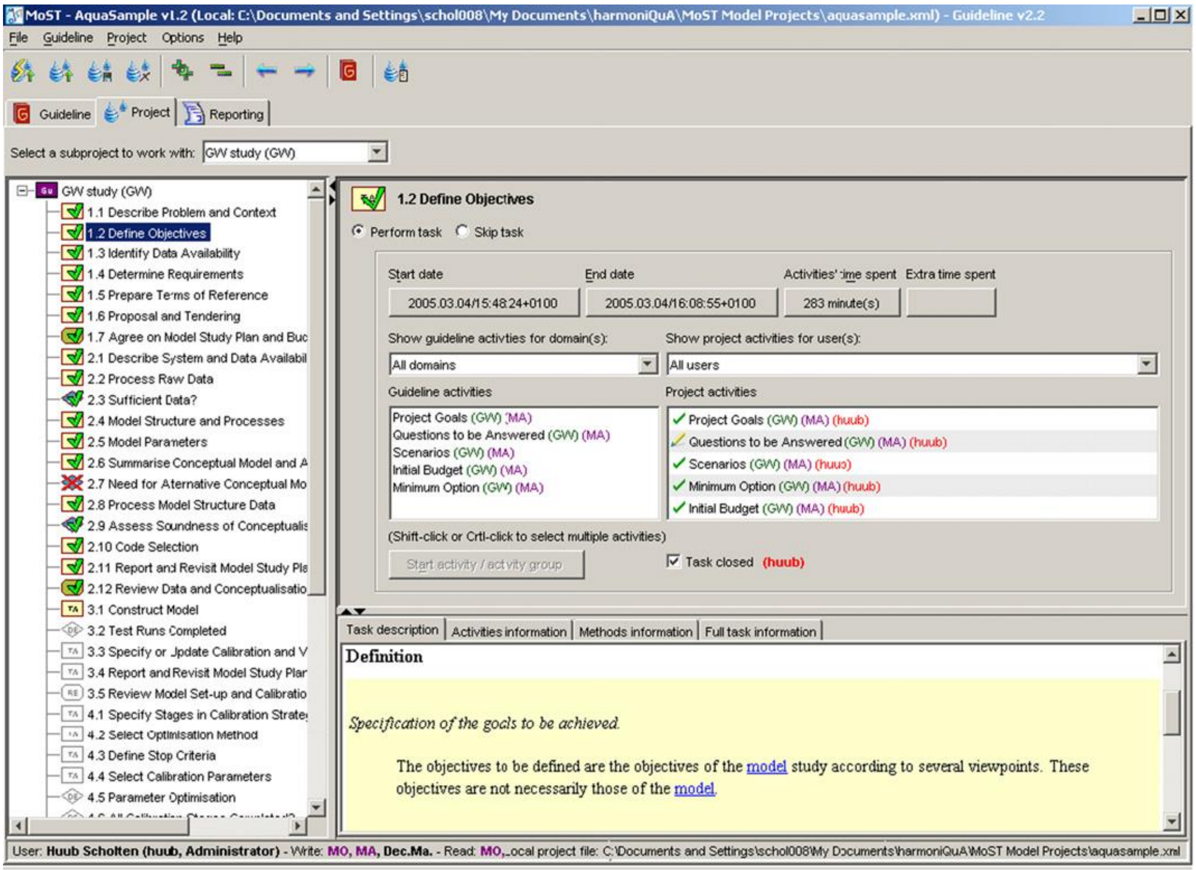


Figure 6-10: A screenshot of MoST (Scholten et al. 2007)

Likewise, the KB contains *process data* only.

In section 6.4.2 we described the PS, KB, and DB as internal systems, and as a result, no SOA roles were assigned to the organisations. The design of MoST as a client-server application enabled the project manager (PM) to setup a shared project log (providing shared PS, KB and DB) and other organisations to use the desktop version of MoST (and be clients to PS). The IT service alignments are, in fact, made based on this reasoning.

6.4.4.3 A Reference Architecture for Modelling Collaboration

Figure 6-11 presents a reference architecture for Integrated Solution Support (I3S). The reference architecture is based on the bootstrap reference architecture (given in Figure 6-4) and the architecture of the I3S provided by Kassahun *et al.* (2008). The architecture includes a large number of IT systems which are described in Table 6-13. Each organisation will not, however, deploy all of the IT systems. Moreover, the web-applications and the persistence layer of an organisation may or may not be shared. Generally, process-related IT systems are provided and shared by the managing organisation and modelling-related IT systems are provided and shared by modelling organisations.

How IT systems and data are shared was defined in AquaStress only informally for lack of modelling abstractions. In contrast, BITA* provides modelling abstractions to do so. In sections 6.4.2 and 6.4.3, and particularly in tables 6.7 and 6.11 we have shown how the distribution IT systems and data across the collaboration network can be modelled, and in tables and 6.8 and 6.12 how the sharing of data can be modelled.

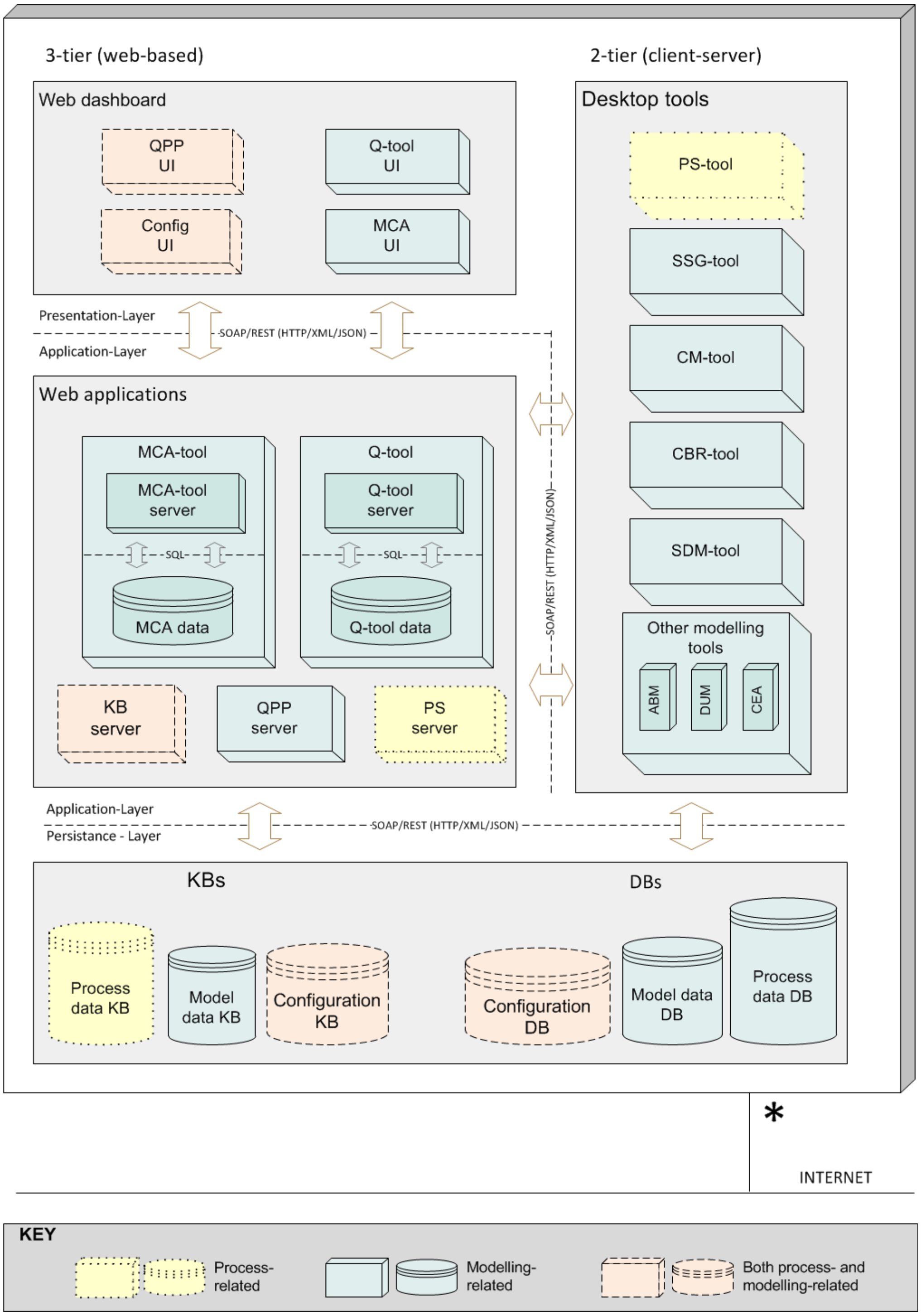


Figure 6-11: Reference architecture for I3S (adapted from Kassahun et al. 2008)

Table 6-13: IT systems of I3S

IT system	*Description*
Process Support (PS)	Using the PS tool the project manager (PM) of a modelling study makes a project plan. The project plan is made by retrieving the entire modelling guideline (modelling collaboration process model) from the process KB and filtering it to the needs of the particular modelling study. The PM shares the project plan using the PS server. Members of participating organisations (including the PM) use a personal PS tool (such as MoST) to retrieve the project plan. The activities of the members are sent to the PS server and registered in the project log (process DB).
Database (DB)	see other descriptions
Knowledge base (KB)	see other descriptions
Questionnaire tool (Q-tool)	A Q-tool is used by modellers (MO) to design questionnaires and by informants (SH) to respond to the questionnaires. A Q-tool is one of the web-based 3-tier IT systems that are encountered in modelling. A Q-tool manages questions and responses in its own DB. Depending on how the questions are formulated a Q-tool DB may or may not be integrated with shared DBs or KBs.
Query processing and presentation tool (QPP-tool)	A QPP-tool is used by all members of participating organisations to visualize process and model data. A QPP-tool retrieves data from shared repositories (DBs and KBs).
Configuration tool (Config-tool)	A Config-tool is used by modellers (MO) to edit the contents of configuration KBs and DBs. An example of a config tool is the KB editor of HarmoniQuA (Kassahun *et al.* 2007), which is used to develop the HarmoniQuA guideline.
Concept mapping tool (CM-tool)	A CM-tool is used to model a "mental map" of a certain reality (Kosko 1986) or to model ontologies (Noy and McGuinness 2001). A CM-tool stores data in process or model KBs.
System dynamics modelling tool (SDM-tool)	A system dynamics model is a semi-quantitative model for studying complex feedback effects of planned interventions. It is a soft systems approach whereby the knowledge of local experts is used to gain insight into feedback mechanisms of systems (Forrester 1994). Generally there are no generic SDM tools. The SDM models of AquaStress are built with SIMILE simulation modelling tool (Vamvakeridou-Lyroudia and Savic 2008). An SDM-tool works with configuration and model repositories.
Multi-criteria analysis tool (MCA-tool)	Multi-criteria analysis is a set of procedures for evaluating alternative decisions involving non-commensurable, conflicting criteria (Triantaphyllou 2000). There are generic MCA tools. The MCA-tool included in I3S of AquaStress is AquaDT, which was developed in the context of the AquaStress project. The MCA-tool works with configuration and model repositories.
Case-based reasoning tool (CBR-tool)	A CBR-tool has been developed as part of I3S as a means of learning by analogy (Griffioen *et al.* 2006). A CBR tool retrieves site specific data from either KBs or DBs.
Strategy Simulation Game (SSG-tool)	Simulation games are computers simulations of a reality for the purpose of gaining insight on how systems work. In role playing games players take on their own or other people's role or behavioural patterns in a real or imaginary context. In strategy games players attempt to manipulate the environment to see the effects of their decisions (Wien *et al.* 2003). In I3S the Splash! strategy simulation game is was included in AquaStress (*ibid*). The SSG-tool works with configuration and model repositories.
Other tools	In addition a number of IT tools, such as *agent-based models* (ABM, Gilbert and Terna 2000, Grimm *et al.* 2006), *data uncertainty models* (DUM, Brown and Heuvelink 2007) and c*ost-effectiveness analysis* (CEA, Berbel *et al.* 2011) are part of the collection IT systems that belong to I3S.

6.5 Discussion

The CatchMod programme in support of Europe's WFD focused on harmonization of modelling processes and modelling IT systems. The harmonization effort led to common glossaries (the HarmoniRiB project, Refsgaard *et al.* 2005), common modelling interface (the HarmonIT project, Moore and Tindall 2005) and common modelling collaboration process models (the HarmoniQuA project, Scholten *et al.* 2007). The harmonization process was challenging for lack of explicit methodology for harmonization.

In this paper we have been able to match the alignment issues to the three classes of alignment concerns of BITA*. We successfully reframed the harmonization of quality assurance guidelines of the HarmoniQuA project as a BP2BP alignment concern, and the development of I3S in the AquaStress project as BP2IT and IT2IT alignment concerns. In literature related to CatchMod projects we were unable to find information about the adopted systematic approach. Ontologies have been used to model processes (Scholten *et al.* 2007, Janssen *et al.* 2009), ensure compatibility of data (Assimacopoulos *et al.* 2009) and define modelling interfaces (Moore and Tindall 2005). But ontologies were used to describe the resulting harmonized process models, IT interfaces or data but not the harmonization process itself. Using the BITA* approach we have been able to derive allocation and alignment matrices and thereby show that the BITA* approach can be used to guide the harmonization process.

Research in modelling collaboration systems has led to a number of new initiatives which were undertaken as research e-infrastructure development initiatives. Examples of such initiatives include the development of scientific workflows, such as Taverna (Wolstencroft *et al.* 2013) and Kepler (Ludäscher *et al.* 2006) which are designed to combine local and remote IT systems into complex analytical workflows. Building Europe's research e-infrastructures, including e-infrastructures for multidisciplinary environmental modelling studies, is part of EU's research agenda (ESFRI 2016). The results of this paper will help the alignment processes that have to be done in order to realize research e-infrastructures.

6.6 Conclusion

In this paper we aimed to identify the alignment concerns in multidisciplinary environmental modelling studies and address them using the BITA* framework. To do so we had to be able to frame the alignment problems encountered in modelling collaboration processes and modelling IT systems as BP2BP, BP2IT, and IT2IT alignment concerns and address them using the BITA* framework.

We have applied the BITA* framework retrospectively to the HarmoniQuA and AquaStress projects. In the HarmoniQuA project a modelling collaboration process model were aligned and an IT system for supporting the model was developed (Kassahun *et al.* 2004, Scholten *et al.* 2007). In the AquaStress various modelling IT systems were aligned to result in an integrated IT system for multidisciplinary environmental modelling (Kassahun *et al.* 2008).

We showed how the alignment modelling would have been done using the BITA* approach. We have achieved the following. (1) We have been able to match the alignment issues to the three classes of alignment concerns of BITA*. We showed how workflow patterns can be used in modelling research collaboration processes. (2) We have been able to derive allocation and alignment matrices and thereby address the alignment concerns. (3) We have shown that BITA* provides systematic approach and the result indicate that it could be applicable to any multidisciplinary projects.

7

GENERAL DISCUSSION

7.1 INTRODUCTION

In this thesis we have explored business-IT alignment problems of multiple collaborating organisations. To cope with these problems we have provided an alignment design framework which we have applied to two different case studies.

The business-IT alignment problems could be identified in both case studies though the case studies come from two distinct application domains. From the results we can derive the following observations.

First, the adoption of a common reference architecture appeared to be necessary to support the business process and IT alignment of collaborating organisations. This has been the case when the collaborating organisations may not initially find a suitable reference architecture that they can use (case study 2) and when they find a usable reference architecture that may not fully meet the requirements of many of the organisations involved (case study 1).

Second, although the concerns of many of the organisations are not addressed by a chosen reference architecture, the collaborating organisations can still find a way to comply with the architecture collectively. Collective compliance was made possible in the first case study because some organisations supported the required business activities and IT on behalf of their collaboration partners. For instance, a transparency system presented in chapter 5 is developed for a third party transparency system provider that provided the required transparency data repository for small food operators (particularly farmers). Deploying the transparency system in a supply chain will make the supply chain EPCIS compliant while many of the food operators do not deploy EPCIS-compliant business processes and IT systems. How the small food operators collaborate with the third party is an alignment concern that requires adaptation of the reference architecture. In this particular case adaption entails defining business collaboration models to address the alignment concerns that were not addressed in the generic reference architecture. Many supply chain transparency systems proposed in the literature (Shanahan *et al.* 2009, Thakur *et al.* 2011, Bruno and Viola 2016) in fact address alignment problems by adapting EPCIS reference architecture.

Third, although BP2BP, BP2IT and IT2IT alignment concerns were initially identified during the first case study, we were able to map the alignment concerns of the second case study to the three types of alignment concerns and address them using BITA*. This is significant because the second case study is about multidisciplinary environmental modelling, a problem domain that is substantially different from that addressed in the first case study, which is about transparency in meat supply chains. This indicates that the alignment framework may be applicable to various other problem domains.

To support solving alignment problems we designed a Business-IT Alignment framework called BITA* and the associated design artefacts. From the case study it became clear that new business collaboration models have to be defined in order to address collaboration concerns that are not addressed in the chosen generic reference architecture. But, each collaborating organisation may have a different view about the required business collaboration models. To align the different views on business collaboration models we provided allocation and alignment models. To design the allocation and alignment models, the business-IT alignment problems were categorized into three types of alignment concerns: Business Process to Business Process (BP2BP), Business Process to IT (BP2IT), and IT to IT

(IT2IT) alignment concerns. For each class of alignment concerns we designed three alignment viewpoints (BP2BP, BP2IT and IT2IT viewpoints), and provided allocation and alignment models that address the specific alignment concerns of each viewpoint. Allocation models represent business collaboration models in a matrix form. Alignment models represent the results of comparing two corresponding allocation models, also in a matrix form. BITA* also provides a systematic approach (a method) for generating alignment matrices. The framework was applied in two case studies in order to verify it, which also led to yet another set of design artefacts. Figure 7-1 summarizes the resulting design artefacts using March and Smith's (1995) categories for design artefacts.

We applied BITA* in two case studies to demonstrate it utility (shown as *implement* RA_1 and *retrospective assessment of* RA_2 in Figure 7-1). The results presented in chapter 3 and 6 show that the framework can indeed help address business-IT alignment problems of multiple collaborating organisations. The framework addressed business-IT alignment problems by helping to derive a more specific reference architecture in which the alignment problems are addressed from a more generic reference architecture in which the alignment problems are not addressed. However, a suitable generic reference architecture may or may not be available, which led to two possible application scenarios of applying BITA*. We have applied the framework in both scenarios as shown in Figure 7-2.

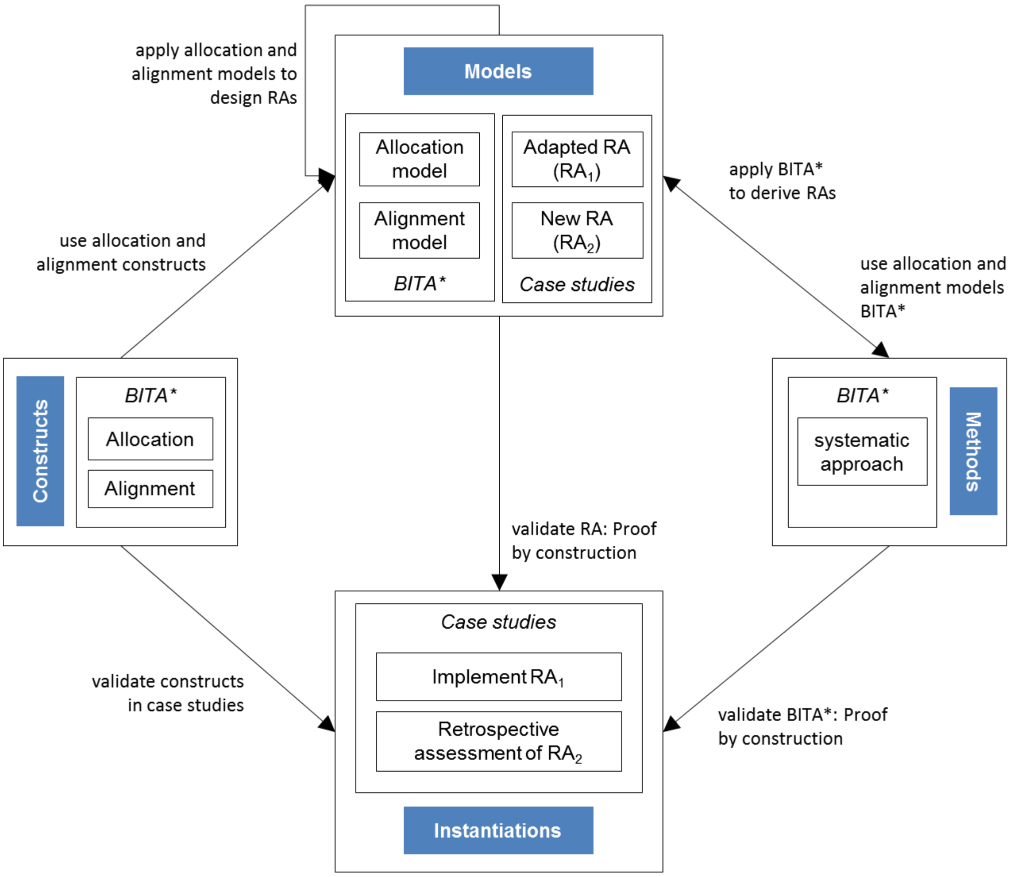


Figure 7-1: The resulting design artefacts of this thesis

(RA: Reference Architecture)

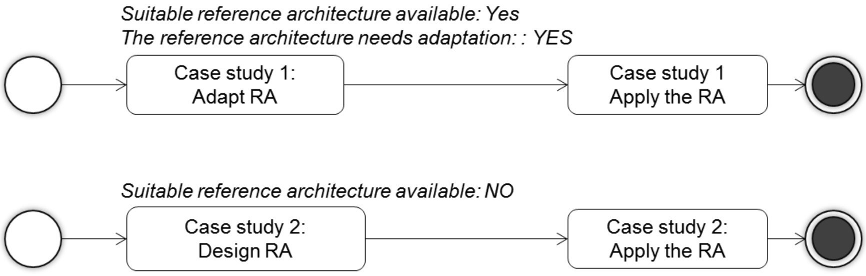


Figure 7-2: The application of BITA in the two case studies of this thesis. (The figure represents two instance of the three possible scenarios depicted in Figure 1-2.)

In the remainder of this chapter we discuss how the research questions were addressed (section 7.2), what was done to mitigate validity threats to the case studies and how the design artefacts are verified (section 7.3), and what the implications of the results of this thesis are (section 7.4).

7.2 ADDRESSING RESEARCH QUESTIONS

7.2.1 Exploring the Alignment Problem

The first research question of this thesis was: *How do alignment problem manifest in multi-organisational collaboration*?

Business-IT alignment problems have generally been addressed within the context of individual organisations, and in that context they are fairly understood. What is explored in this thesis is how business-IT alignment problems manifest themselves from the perspective of multi-organisational collaboration, which is poorly understood. We explored alignment problems in two case studies. In the following we summarize the results.

7.2.1.1 Case study 1: Meat Supply Chains

The first case study has two major purposes: (1) to elaborate how business-IT alignment problems manifest themselves when multiple organisations need to collaborate, and (2) to help derive and test business-IT alignment framework. In this section we discuss the first purpose.

The case study demonstrated how business-IT alignment problems occur in meat supply chain transparency systems. If fact, this case study provided an ideal case for demonstrating alignment problems when multiple and diverse organisations collaborate. This is because an essential aspect of transparency, which is the ability to track and trace product items across the supply chain whenever and wherever required, requires the collaboration of many and diverse actors (see chapter 4 for the detailed description). Meat supply chains involve multiple and diverse organisations, including farmers, slaughterhouses, diverse types of meat processors, logistic operators, food regulators and retailers. These organisations not only differ in their roles but also in their sizes. In addition each role may be performed by many organizations (e.g. many farmers and many meat processors). These differences impact their capabilities in supporting required business activities and providing required IT systems.

Besides, food operators in meat supply chains are subject to different and partly contradictory regulatory mandates. For instance, we have shown that in Europe part of meat supply chains that involves farmers and slaughterhouses is subject to the regulation Reg. N^{o} 1760/2000 (EC

2000) which requires a nationwide identification and registry system for animals. The other half involving slaughterhouses and other meat processors is subject to the regulation Reg. N° 178/2002 (EC 2002), also called the General Food Law, which mandates the *one-step-back/one-step-forward* principle. In the former transparency data are managed by a centralized transparency data repository, in the later transparency data are managed by the food operators themselves. The two approaches demand different business collaboration process models and different types of IT support.

Transparency is a major business and societal concern in general and, as a result, there is global consortium (with member organisations in all major countries) that designs global standards for supply chain transparency. The standard that defines a generic reference architecture for transparency is EPCIS (GS1 EPCglobal 2014). EPCIS provides a fixed set of IT interfaces and data models that organisations have to comply with in order for them to be part of supply chain wide transparency systems. In terms of the EPCIS reference architecture all organisations are considered to be similar in their ability to comply with the standard. However, many food operators in meat supply chains cannot comply with the EPCIS reference architecture for various reasons. We identified five different types of organisations who have their own views how a reference architecture for a chain-wide transparency system should be realized. Besides identifying the different alignment concerns, the case study showed how the EPCIS reference architecture can be adapted by aligning the different views (see chapters 4 and 5).

7.2.1.2 Case study 2: Enviromental Modelling

Business-IT alignment problems are also encountered in other application domains. The second case study (see chapter 6) is a retrospective analysis of two alignment projects in multidisciplinary environmental modelling domain.

This case study differs from the first one in two major ways. First, there was no suitable generic reference architecture to start with. Second, environmental modelling studies (and projects in general) are managed using project planning approaches but not using business process modelling approaches. To identify business-IT alignment problems with the help a generic reference architecture we defined an initial, bootstrap, reference architecture; and we also showed that the business process modelling approach is, in fact, already widely used to manage modelling studies, though conceptualized differently, such as quality assurance procedures. We were then able to elaborate alignment problems encountered in multidisciplinary modelling as business process to IT alignment problems. We showed how alignment helped develop the quality assurance guideline in the HarmoniQuA project, the integrated modelling support system in the AquaStress project.

7.2.2 Deriving the BITA Framework*

The second research question of this thesis was: *What should a business-IT alignment framework constitute*?

The BITA* framework is the result of considering *aligning* as a design artefact and as a major design concern. To derive the framework we introduced two design constructs: *allocation* and *alignment*. Allocation and alignment are the means with which we compare models that are designed as BPMN or SOA-related modelling abstractions. We also introduced the concept of *business collaboration model* to model collaborations. We defined three types of business collaboration models. A business collaboration process model describes a business process that spans across a collaboration network. A business collaboration IT model describes the distributed IT system that underlies business collaboration processes. We also introduced a business collaboration process-IT model that represents the relationship between the process and IT models.

To understand the BITA* approach, it is important to realize that there is no central authority that designs business collaboration models. We showed that it is reasonable to assume that each collaboration organisation has a particular view on what is required to realize business collaboration across the collaboration network and has its own business collaboration models for it (albeit implicit). The corresponding business collaboration models from the different collaborating organisation have to be aligned in order to define a reference architecture that the organisations will agree with.

In BITA* we defined allocation and alignment models and the associated systematic approach for aligning the business collaboration models of the different collaborating organisations. The models are defined in design viewpoints. A viewpoint addresses design concerns of a particular type of stakeholders by providing them with explicit modelling abstractions. The modelling abstraction can be entirely new but often are built on top of already existing modelling abstractions from related design viewpoints (Clements *et al.* 2010, ISO/IEC/IEEE 2011). In BITA* we identified three types of stakeholders and designed three viewpoints that address their alignment concerns. The first viewpoint addresses the concerns of business analysts. Business analysts deal with the alignment of the business processes of their organisation with those of the collaboration partners. We call their concerns BP2BP alignment concerns and the associated viewpoint BP2BP viewpoint. The second viewpoint addresses the concerns of software architects. Software architects deal with the integration of the IT systems of their organisation with those of the collaboration partners. We call their concerns IT2IT alignment concerns and the associated viewpoint IT2IT viewpoint. The third viewpoint addresses the concerns of an interdisciplinary team of business analysts and software architects. Such a team deals with the alignment of business processes with the underlying distributed IT system. We call their concerns BP2IT (or IT2BP) alignment concerns and the associated viewpoint BP2IT viewpoint.

In BITA* we encountered three major design problems. (1) Alignment requires comparing business collaboration models. But, comparing models is generally hard for several reasons including differences in the notation and modelling abstractions used, differences in the abstraction levels chosen, and complexity of models (van der Aalst *et al.* 2006, Dijkman *et al.* 2008, Weidlich *et al.* 2011, Ahmad *et al.* 2012). (2) Even if models would be comparable, the required pairwise comparison rises rapidly as the number of participating organisations increases. (3) In practice, many of the collaborating organisations cannot produce explicit models which can be used in the alignment process. To address these problems we have provided two solutions. (1) We used allocation models to represent business collaboration models as matrices and tables with the help of workflow patterns. The tabular representation enabled us to compare two corresponding business collaboration models and make explicit statement about their alignments. Workflow patterns also enabled us to be more explicit about the abstraction levels used. (2) We used reference models so that the collaborating organisations do not have to produce explicit business collaboration models of their own. We represented the organisations’ perspectives about business collaboration using reference models and alignment attributes. We defined convergence, partial convergence, divergence, partial divergence, and absence alignment attributes to model alignments.

7.2.3 Applying the Alignment Framework

The third research question of this thesis was: *How to apply business-IT alignment framework to address collaboration concerns?*

The ultimate goal of BITA* is to help build a distributed IT system that supports collaboration across multiple organisations. Verifying BITA* requires realizing and testing collaboration systems. It is important to realize, though, that BITA* is an alignment framework, and not a software system development framework. Therefore, to truly validate BITA* a multi-step process was required. We used four steps to validate BITA*, which are: (1) deriving a reference architecture with the help of BITA*; (2) deriving one or more concrete architectures and business collaboration processes from the reference architecture; (3) building a distributed IT system (a collaboration system) using the concrete architectures; and finally, (4) demonstrating the distributed IT system by executing the business collaboration processes to address specific collaboration problems.

We have derived two reference architectures using BITA*, which are described in chapters 3, 5 and 6 and in previous publication (Scholten *et al.* 2007, Kassahun *et al.* 2008). The first reference architecture has been used to derive a concrete architecture and the corresponding collaboration system as described in chapters 3 and 5. We have also demonstrated the system in the same chapter. To fully demonstrate its utility the system has to be deployed across a collaboration network. The demonstration presented in chapter 5 is currently one of the pilot studies of the newly launched Internet of Food and Farm (IoF2020) project[1]. The second reference architecture is a retrospective analysis of a collaboration system that was already implemented and tested (Scholten *et al.* 2007, Kassahun *et al.* 2008, Assimacopoulos *et al.* 2009, Inman *et al.* 2011).

7.3 EVALUATIONS

7.3.1 Validity of Case Study Results

Case study research is susceptible to various validity threats, which have to be addressed for the results to be acceptable. The major threats to validity are *construct validity*, *internal validity*, *external validity*, and *reliability* (Yin 2003). Mitigating strategies for threats of validity in software engineering related case studies include: *prolonged involvement* to enhance shared understanding of concepts, *triangulation* of data from different informants to capture and explain contradictory views, *peer debriefing* to avoid bias of sole investigator, *member checking* by informants of the study to overcome misunderstanding, *negative case analysis* to formulate alternative explanations and theories to improve the analysis information, and *audit trail* to be able to present chain of evidence (Runeson *et al.* 2012). In the following we describe how we applied these mitigation strategies in our case studies.

Construct Validity

Construct validity entails that the constructs are measured or interpreted correctly (Yin 2003). Constructs are conceptualizations, often abstract, and as such, not directly usable. The

[1] http://iof2020.eu/

allocation and *alignment* constructs of this thesis are very good examples of abstract constructs. Constructs are operationalized by either associating them with measurement variables or by qualifying them with adequate explanations (Yin 2003). We made allocation and alignment measurable by providing explicit allocation and alignment models. *Allocation* is associated with the binary *true* or *false* (*yes* or *no*) values and *alignment* is associated with one of the five possible alignment attributes: *convergent*, *divergent*, *absent*, *partially divergent*, and *partially absent*. The alignment attributes are themselves constructs which are qualified through adequate explanation in reflexion modelling literature. *Business collaboration model* is yet another construct introduced in this thesis that is operationalized using allocation modelling.

Many more constructs are used in this thesis. They can be categorized as actor, organisation, IT, activity, or a type derived from any combination of these (Sjøberg *et al.* 2008). The constructs we used are derived from BPMN, SOA and workflow pattern modelling approaches. These constructs are widely used and validated in literature.

To enhance construct validity Yin suggests three mitigation strategies: using multiple sources of evidence, establishing a chain of evidence, and letting informants review the constructs. In the first case study, from which the constructs were derived, we used multiple sources of evidence. We used informants from two meat processors, farmers, GS1, Orgainvent, EHI, Global G.A.P., QS, and two supermarket chains. The other two mitigation strategies turned out difficult to apply in our case studies because of the informal and iterative nature of the design process followed. Instead of a separate informant review step we actively involved informants in various occasions and various ways throughout the iterative process (see chapters 4 and 5).

Internal Validity

Internal validity is about cause-and-effect relationships among variables, a concern that has to be dealt with in explanatory case studies. It ensures that the observed effects are only due to the identified causes (Yin 2003). When internal validity is an issue, it is often confused with construct validity. But, the two are distinctly different; the former is about the accounting for all alternative *cause variables* while the latter is about the measurement or the interpretation of both *cause* and *effect variables* (Straub *et al.* 2004). Internal validity is enhanced by showing: (1) there is a clear *correlation* between cause and effect variables, (2) the cause variables are *antecedent* to effect variables, and (3) there are no *confounding variables*. Strategies to mitigating internal validity threats are pattern matching and explanation building (Yin 2003). Pattern matching is suitable when there are quantitative data, which is not the case in design science related research, which this thesis represents; therefore, we provided mainly explanations to support internal validity.

Ensuring internal validity in crucial for both case studies. The primary claim of design science research is create design artefacts (effects) that address the requirements for the design (causes). In this case study once the requirements were identified it took a couple of years before the required design artefacts were realized and tested. This causes a concern that the resulting designs may not have been the only reason that the requirements are fulfilled. The artefacts we developed seemed to fulfil the requirements but it is still important to make sure that the requirements were fulfilled only because of the artefacts not confounded by other variables. Other variables could be the deployment of other design artefacts or the pressure to make the design project a success—while better alternative solutions are available. The later could, for instance, lead to the implementation to drift substantially from the original designs, and thus the systems tested do not reflect the designs.

To ensure internal validity we had a prolonged encounter with informants and were able to formulate detailed explanations of the requirements, designs and systems. We also applied triangulation and negative case analysis to justify requirements and designs. For instance, in the first case study we involved informants from Global G.A.P to solicit negative case analysis. Global G.A.P uses a different approach to transparency which is based on certification and inspection, while our approach (based on the EPCIS standard) is based on real-time data capturing and sharing. Also, the fact that the results of the first case are considered as one of the large scale trails to be conducted in Europe under the Internet of Food and Farm (IoF2020) project indicates that: (1) the designs (effects) address the requirements (causes), and (2) the requirements are still valid and the designs are still relevant.

External Validity

External validity entails that claims of generality are justified, which means that the results will also apply to different contexts. Strategies for mitigating external validity threats are either using theories in addition to a case study, using multiple case studies, or both (Yin 2003). We enhanced the external validity of the study by replicating the first case study in a different application area. Replication of a large design-related case study in a new project is, however, problematic because finding a real-life large scale collaborative project is difficult to find or organise. We have, therefore, used a retrospective case study.

Reliability

Reliability in engineering case studies entails that the results can be replicated within the same context. In data intensive case studies reliability is enhanced using mitigation strategies based on statistics, such as dividing the available data in to two parts and using one part for calibration and the other for validation (Straub *et al.* 2004)—an option not suitable for our case studies. Instead, we have extensively consulted with peers within and outside the case studies (peer debriefing) and consulted target users and companies on the requirements and proposed solution (member checking) to mitigate reliability issues.

7.3.2 Evaluation of Design Artefacts

A review of design science literature shows that a design science research process may constitute the following steps: (1) identify and motivate the problem, (2) define requirements, (3) design and instantiate, (4) demonstrate the solution, (5) evaluate, and (6) communicate (Peffers *et al.* 2008). We have mentioned earlier that we used case study research for the first two steps in designing the BITA* framework. The last step, communication, has been done through various means, including web site, presentation a number scientific conferences and publication in scientific journals.

The three intermediate steps did not follow the sequential process described by Peffers *et al.* This was also to some extent true about the first two steps. However, whether the first two steps were done sequentially or iteratively did not have much impact on evaluation. The iterative nature of the three next steps did, however, has impact on the evaluation of design artefacts.

According to the above six-steps process, once the design artefacts are developed (designed and instantiated), they should be demonstrated, and then evaluated. When we consider BITA* as a design artefact, this approach is, however, not that straightforward for at least to major reasons.

First, as described previously, applying the BITA* framework to help realize a collaboration system requires multiple steps, and each one of them can be considered as a design science research. The steps include designing a reference architecture, applying the reference architecture to derive concrete architectures, using the concrete architectures to realize IT systems, and using the realized IT systems. This resulted in a cascade of design artefacts and, as a result, it is not clear to which artefacts the demonstration and evaluation steps apply. We consider that the steps of demonstration and evaluation apply to the final, realized, IT systems. In this respect the evaluation of the final IT systems are available as published articles (Scholten *et al.* 2007, Assimacopoulos *et al.* 2009, Inman *et al.* 2011). The intermediate design artefacts (architecture and software designs) are validated by actually instantiating them, or to use Peffers *et al.*'s terminology through "proof by construction" (Nunamaker *et al.* 1990, Hevner *et al.* 2004).

Second, the design artefacts are not developed sequentially as described above but iteratively. As a result the boundaries of the first five steps are not crisp. Evaluations were made routinely but to identify new requirements for the next round of development. The stakeholders were actively involved and the design artefacts were continuously evaluated by them. In fact, an iterative and agile development methodology (Fowler and Highsmith 2001) was adopted throughout the FIspace project. The first case study was conducted in the context of that project and the resulting design artefacts were developed iteratively. In iterative approaches evaluations are integral part of the design and instantiation process (Sein *et al.* 2011). In agile methodologies the aim is to ensure that valuable design artefacts are available even if the project is suddenly stopped, for instance, due to budget restrictions (Hunt 2006). As a consequence, evaluations were largely used to derive or enhance requirements instead of measuring the degree of success.

The reference architecture of the first case study, which is the main design artefact of the case study, was validated to an extent by the concrete implementation of a transparency system presented in chapter 5. To fully validate the reference architecture we have to derive multiple concrete architectures for different organisations, realize the corresponding IT systems, deploy the IT systems, and evaluate the resulting business collaboration processes and distributed IT. We are currently undertaking such a large scale demonstration and evaluation within the IoF2020[2] project.

7.4 Implications of this Thesis

7.4.1 Theoretical Implications

In this thesis we have derived *allocation* and *alignment* design constructs and the associated models to make model comparisons possible. We thereby considered each organisation's business collaboration models for comparison. This approach is significantly different from the approaches that have been suggested in literature. In literature the general approach to

[2] http://cordis.europa.eu/project/rcn/206761_en.html

compliance and alignment, often applied to business process models (Kunze *et al.* 2011, Weidlich *et al.* 2011, Rosa *et al.* 2013) does not consider business collaboration models explicitly. We believe our approach will provide a novel approach to addressing alignment when multiple organisations are involved.

We have provided three scenarios illustrating how alignment and reference architectures are related in Figure 1-2. The most widely used scenario in practice is scenario ①, in which an existing reference architecture is used to derive (or adapt) concrete architectures so that the concrete architectures are aligned. The use of alignment modelling depicted as scenarios ② and ③ is, however, novel, and will help future alignment projects in which generic reference architecture, such as EDI (Mukhopadhyay *et al.* 1995), SCOR (SCC 2012) and EPCIS (EPCglobal 2014) will be adapted for specific purposes and domains.

Originally workflow patterns were meant to be used for investigating the suitability of business process modelling abstractions, and for assessing workflow and "process-aware" systems (van der Aalst and ter Hofstede 2011). We have shown how workflow patterns can be used for making models comparable and help design reference architectures. The application of workflow patterns for the purpose of allocation modelling is also novel, and in fact, this thesis showed that workflow patterns are essential enablers of business-IT alignment.

7.4.2 Practical Implications

The results of this thesis have several practical implications, which can be grouped into generic and sector specific.

The generic practical implication is related to methodologies to alignment. Reference architectures are probably the primary means of realizing integration across large collaboration networks. Alignment, though not explicitly stated, has been one of the required steps in deriving sector-specific reference architectures from generic ones. In the agri-food sector, for instance, many reference architectures have been developed by adapting and extending existing generic reference architectures, such as EPCIS, EDI, ISA-95(Scholten 2007) and SCOR (Verdouw *et al.* 2010, Medini and Bourey 2012, Bruno and Viola 2016, Kidane and Kim 2016, Jung *et al.* 2017, Kruize 2017). The reference architectures are derived by considering many different use case scenarios. The resulting reference architectures are presented using modelling abstractions, such as BPMN, ArchiMate (The Open Group 2013), and UML. The alignment process adopted is usually not explicitly stated, but involves some form of alignment process. The lack of explicit modelling abstractions for alignment and the lack of systematic approaches to alignment constitute major methodological gap. The BITA* framework fills this gap.

The sector specific implications are the reference architectures and the practical prototype collaboration systems that are derived from them. The meat transparency case study has resulted in a transparency system that is implemented by an SME (Small and Medium-sized Enterprise) and has recently received funding for large scale rollout. The results of this thesis have a practical implication for the involved SME and for the businesses involved in the large scale rollout. The multidisciplinary modelling case study has revisited former harmonization projects in a systematic way. We have shown how the BITA* approach may provide an efficient method for harmonization. Supporting multidisciplinary environmental modelling studies is part of EU's long standing support for building Europe's research e-infrastructures

(ESFRI 2016). The results of this study can help the harmonization projects that are being undertaken as part of the research e-infrastructure development effort.

7.4.3 Future Research

In this thesis we have presented a novel approach to business-IT alignment for multiple collaborating organisations. We applied explicit models, generic reference architectures, and workflow patterns in modelling alignment. We have addressed many of the issues we have come across; for instance, how to make models comparable so that we can make explicit statements about the state of alignment. But, the results also raised new research questions that hint towards new avenues for further research. In the following we point out some of them.

First, we used BPMN for representing business collaboration processes. But there are many other modelling abstractions for representing collaboration. For instance, a declarative modelling technique called Guard-Stage-Milestone (GSM, Hull *et al.* 2011) was used in the FIspace project. The declarative modelling approach used GSM differs significantly from the imperative approach used in BPMN. Future research may consider how to incorporate other business process modelling abstractions in allocation modelling.

Second, we have used SOA service models to represent a distributed IT system (in terms of IT systems, IT services, data objects and the roles organisations take in providing or using them). However, there are many aspects to modelling a distributed IT system. Some of these aspects are discussed, for instance, as patterns for distributed computing (Buschmann *et al.* 2007) and cloud computing (Armbrust *et al.* 2010). Future research may consider how to incorporate various other aspects of modelling a distributed IT in allocation modelling.

Third, though workflow patterns were an essential part of allocation models, they were not originally developed for representing business collaboration models. Patterns are generally described using attributes such as *description*, *example*, *motivation* and *context*. The available workflow patterns represent business process patterns found within an organisation. As a result, the descriptions, examples, motivations and contexts refer to patterns found within an organisation. Further research is required to adapting existing workflow patterns to reflect business collaboration cases. Also, future research may provide new workflow patterns for business collaboration models.

Fourth, we defined two allocations attributes: *true* (*yes*, allocated) or *false* (*no*, not allocated); and we defined five alignment attributes: *convergence*, *partial convergence*, *divergence*, *partial divergence* and *absence*. But, many times it is very difficult to make a clear *yes* and a clear *no* statement about allocation. It is not uncommon that an activity, data or IT service supports some of the required elements and some not. If the allocation is somewhere in between a *yes* and a *no*, the problem propagates to the alignment modelling, and the given five alignment attributes will not be enough. A possible solution is to adopt the fuzzy or soft systems approaches (Zadeh 1994, Checkland *et al.* 2010) to deal with uncertainties. Further elaboration of this problem is required in future research.

Fifth, we have provided BITA* and applied it in two case studies. But there are various application areas where business-IT alignment across collaboration networks is a concern. For instance, the supply chain literature identifies other types of collaboration concerns besides transparency, such as, inventory management, logistics management and efficient processing of consumer response (Holweg *et al.* 2005, Attaran and Attaran 2007). Future research can show how to apply the framework in other application domains.

Sixth, future research can provide a design support system for further assisting business analysts and software architects in modelling allocations and alignments.

REFERENCES

Ackoff, R. L. (1971). "Towards a system of systems concepts." *Management science*. **17**(11): 661-671.

Ahmad, A., P. Jamshidi and C. Pahl (2012). Graph-Based Pattern Identification from Architecture Change Logs. *Advanced Information Systems Engineering Workshops: CAiSE 2012 International Workshops, Gdańsk, Poland, June 25-26, 2012. Proceedings*. M. Bajec and J. Eder. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 200-213.

Albersmeier, F., H. Schulze, G. Jahn and A. Spiller (2009). "The reliability of third-party certification in the food chain: From checklists to risk-oriented auditing." *Food Control*. **20**(10): 927-935.

Angelov, S., P. Grefen and D. Greefhorst (2012). "A framework for analysis and design of software reference architectures." *Information and Software Technology*. **54**(4): 417-431.

APICS. (2017). "American Production and Inventory Control Society (APICS)."   Retrieved January 17, 2017, from http://www.apics.org/about/overview.

AquaStress. (2006). "The project AquaStress."   Retrieved January 7, 2017, from http://www.aquastress.net/.

Argent, R. M. (2004). "An overview of model integration for environmental applications--components, frameworks and semantics." *Environmental Modelling & Software*. **19**(3): 219-234.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia (2010). "A view of cloud computing." *Commun. ACM*. **53**(4): 50-58.

Arnold, G. E., W. J. d. Lange and M. W. Blind (2005). "The concerted action Harmoni-CA: Facilitating the dialogue and bridging the gap between research and the WFD implementation." *Environmental Science and Policy*. **8**(3): 213-218.

Assimacopoulos, D., S. Zaccolo, R. Maia, I. Glekas, Z. L. Chabaâne, A. Hammani, I. Ribarova, E. b. D. yńska and HenkWolters (2009). Water Stress Mitigation: The AquaStress Case Studies. Greece.

Athanasiadis, I. N., A.-E. Rizzoli, S. Janssen, E. Andersen, F. Villa, F. Sartori, M. Á. Sicilia and N. Manouselis (2009). Ontology for Seamless Integration of Agricultural Data and Models
Metadata and Semantic Research, Springer Berlin Heidelberg. **46:** 282-293.

Attaran, M. and S. Attaran (2007). "Collaborative supply chain management: The most promising practice for building efficient and sustainable supply chains." *Business Process Management Journal*. **13**(3): 390-404.

Aversano, L., C. Grasso and M. Tortorella (2016). "Managing the alignment between business processes and software systems." *Information and Software Technology*. **72**: 171-188.

Barmpounakis, S., A. Kaloxylos, A. Groumas, L. Katsikas, V. Sarris, K. Dimtsa, F. Fournier, E. Antoniou, N. Alonistioti and S. Wolfert (2015). "Management and control applications in Agriculture domain via a Future Internet Business-to-Business platform." *Information Processing in Agriculture*. **2**(1): 51-63.

Barratt, M. (2004). "Unveiling Enablers and Inhibitors of Collaborative Planning." *The International Journal of Logistics Management*. **15**(1): 73-90.

Barry, D. K. (2003). *Web services, service-oriented architectures, and cloud computing*, Morgan Kaufmann.

Bass, L., P. Clements and R. Kazman (2003). *Software architecture in practice*, Addison-Wesley Professional.

Bass, L., P. Clements and R. Kazman (2012). *Software Architecture in Practice*, Addison-Wesley Professional.

BDMF (2000). Protocols for Water and Environmental Modeling, Bay-Delta Modeling Forum. Ad hoc Modeling Protocols Committee. www.sfei.org/modelingforum/.

Becker, M. and R. Laue (2012). "A comparative survey of business process similarity measures." *Computers in Industry*. **63**(2): 148-167.

Berbel, J., J. Martin-Ortega and P. Mesa (2011). "A Cost-Effectiveness Analysis of Water-Saving Measures for the Water Framework Directive: the Case of the Guadalquivir River Basin in Southern Spain." *Water Resources Management*. **25**(2): 623-640.

Bertolini, M., M. Bevilacqua and R. Massini (2006). "FMECA approach to product traceability in the food industry." *Food Control*. **17**(2): 137-145.

Bhatt, T., G. Buckley and J. C. McEntire (2012). "Proceedings of the August 2011 Traceability Research Summit." *Journal of Food Science*.

Blackstone, J. H. and J. F. Cox (2005). *APICS Dictionary*, American Production & Inventory Control Society, Incorporated.

Boardman, J. and B. Sauser (2006). *System of Systems-the meaning of of*. IEEE/SMC International Conference on System of Systems Engineering, IEEE.

Bosona, T. and G. Gebresenbet (2013). "Food traceability as an integral part of logistics management in food and agricultural supply chain." *Food Control*. **33**(1): 32-48.

Bouguettaya, A., Q. Z. Sheng, F. Daniel and C. Pautasso (2014). RESTful Web Services: Principles, Patterns, Emerging Technologies. *Web Services Foundations*, Springer New York**:** 31-51.

Bowling, M. B., D. L. Pendell, D. L. Morris, Y. Yoon, K. Katoh, K. E. Belk and G. C. Smith (2008). "Review: Identification and Traceability of Cattle in Selected Countries Outside of North America." *The Professional Animal Scientist*. **24**(4): 287-294.

Brewster, C., S. Wolfert and H. Sundmaeker (2012). *Identifying the ICT challenges of the Agri-Food sector to define the Architectural Requirements for a Future Internet Core Platform*. eChallenges e-2012: IIMC International Information Management Corporation, Lisbon, Portugal.

Brom, F. A. (2000). "Food, Consumer Concerns, and Trust: Food Ethics for a Globalizing Market." **12**(2): 127-139.

Brown, J. D. and G. B. M. Heuvelink (2007). "The Data Uncertainty Engine (DUE): A software tool for assessing and simulating uncertain environmental variables." *Computers & Geosciences*. **33**(2): 172-190.

Bruno, G. and V. Viola (2016). A Collaborative Architecture for Supply Chain Transparency Based on EPCIS Standard and MongoDB. *Collaboration in a Hyperconnected World: 17th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2016, Porto, Portugal, October 3-5, 2016, Proceedings*. H. Afsarmanesh, L. M. Camarinha-Matos and A. Lucas Soares. Cham, Springer International Publishing**:** 599-607.

Buschmann, F., K. Henney and D. C. Schmidt (2007). *Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing*, John wiley & sons.

Buyya, R., C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic (2009). "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation Computer Systems*. **25**(6): 599-616.

Cañas, A. J., G. Hill, R. Carff, N. Suri, J. Lott, T. Eskridge, G. Gómez, M. Arroyo and R. Carvajal (2004). *CmapTools: A knowledge modeling and sharing environment*. Concept maps: Theory, methodology, technology. Proceedings of the first international conference on concept mapping.

Castrucci, M., M. Cecchi, F. Delli Priscoli, L. Fogliati, P. Garino and V. Suraci (2011). *Key concepts for the Future Internet architecture*. Future Network & Mobile Summit (FutureNetw), 2011.

Chan, E. Y. and H. B. Reich (2007). "IT alignment: what have we learned?" *Journal of Information Technology*. **22**(4): 297-315.

Chau, P. Y. K. and K. Y. Tam (2000). "Organizational adoption of open systems: a 'technology-push, need-pull' perspective." *Information & Management*. **37**(5): 229-239.

Checkland, P., J. Poulter, M. Reynolds and S. Holwell (2010). Systems Approaches to Managing Change: A Practical Guide, Springer London**:** 191-242.

Checkland, P. and M. Winter (2005). "Process and content: two ways of using SSM." **57**(12): 1435-1441.

Chen, D., G. Doumeingts and F. Vernadat (2008). "Architectures for enterprise integration and interoperability: Past, present and future." *Computers in Industry*. **59**(7): 647-659.

Chen, H.-M., R. Kazman and A. Garg (2005). "BITAM: An engineering-principled method for managing misalignments between business and IT architectures." *Science of Computer Programming*. **57**(1): 5-26.

Cheng-Leong, A., K. Li Pheng and G. R. Keng Leng (1999). "IDEF*: A comprehensive modelling methodology for the development of manufacturing enterprise systems." *International Journal of Production Research*. **37**(17): 3839-3858.

Chinosi, M. and A. Trombetta (2012). "BPMN: An introduction to the standard." *Computer Standards & Interfaces*. **34**(1): 124-134.

Chung, L. and J. C. P. Leite (2009). On Non-Functional Requirements in Software Engineering. *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, Springer-Verlag**:** 363-379.

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord and J. Stafford (2010). *Documenting software architectures: views and beyond*, Addison-Wesley.

Cloutier, R., G. Muller, D. Verma, R. Nilchiani, E. Hole and M. Bone (2010). "The Concept of Reference Architectures." *Systems Engineering*. **13**(1): 14-27.

Codd, E. F. (1970). "A relational model of data for large shared data banks." *Communications of the ACM*. **13**(6): 377-387.

Collee, J. G. and R. Bradley (1997). "BSE: a decade on—part I." *The Lancet*. **349**(9052): 636-641.

Crasso, M., A. Zunino and M. Campo (2013). A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. *Innovations in Database Design, Web Applications, and Information Systems Management*, IGI Global**:** 107-138.

Cummins, F. A. (2015). BPM Meets SOA: A New Era in Business Design. *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*. J. vom Brocke and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 531-555.

Daclin, N., S. M. Daclin, V. Chapurlat and B. Vallespir (2016). "Writing and verifying interoperability requirements: Application to collaborative processes." *Computers in Industry*. **82**: 1-18.

Dahmann, J. S., G. Rebovich Jr and J. A. Lane (2008). Systems engineering for capabilities, DTIC Document.

Davenport, T. H. (2005). "The coming commoditization of processes." *Harvard Business Review*. **83**(6): 100-108.

Davenport, T. H. and J. E. Short (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." *Sloan Management Review*. **31**(4): 11-27.

Davenport, T. H. and D. B. Stoddard (1994). "Reengineering: Business Change of Mythic Proportions?" *MIS Quarterly*. **18**(2): 121-127.

Dijkman, R. M., M. Dumas and C. Ouyang (2007). "Formal semantics and analysis of BPMN process models using Petri nets."

Dijkman, R. M., M. Dumas and C. Ouyang (2008). "Semantics and analysis of business process models in BPMN." *Information and Software Technology*. **50**(12): 1281-1294.

Dijkman, R. M., B. F. van Dongen, M. Dumas, L. García-Bañuelos, M. Kunze, H. Leopold, J. Mendling, R. Uba, M. Weidlich, M. Weske and Z. Yan (2013). A Short Survey on Process Model Similarity. *Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE*. J. Bubenko, J. Krogstie, O. Pastor et al. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 421-427.

Donnelly, K. A.-M., K. M. Karlsen and P. Olsen (2009). "The importance of transformations for traceability – A case study of lamb and lamb products." *Meat Science*. **83**(1): 68-73.

Duffy, R., A. Fearne and V. Healing (2005). "Reconnection in the UK food chain: Bridging the communication gap between food producers and consumers." *British Food Journal*. **107**(1): 17-33.

Dumas, M., M. La Rosa, J. Mendling and H. Reijers (2013). Introduction to Business Process Management. *Fundamentals of Business Process Management*, Springer Berlin Heidelberg**:** 1-31.

Easterbrook, S., J. Singer, M.-A. Storey and D. Damian (2008). Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering*. F. Shull, J. Singer and D. I. K. Sjøberg. London, Springer London**:** 285-311.

Ebling, M. and R. Cáceres (2010). "Bar Codes Everywhere You Look." *Pervasive Computing, IEEE*. **9**(2): 4-5.

EC (2000). "Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for Community action in the field of water policy." *Official Journal of the European Communities*. **327**: 1-73.

EC (2000). "Regulation (EC) No 1760/2000 of the European Parliament and of the Council of 17 July 2000 establishing a system for the identification and registration of bovine animals and regarding the labelling of beef and beef productsand repealing Council Regulation (EC) No 820/97." *Official Journal of the European Communities*(L 204): 1-10.

EC (2002). "Regulation (EC) No 178/2002 of the European Parliament and of the Council of 28 January 2002 laying down the general principles and requirements of food law, establishing the European Food Safety Authority and laying down procedures in matters of food safety." *Official Journal of the European Communities*. **L 31**(1): 1-24.

EC (2003). Common Implementation Strategy for the Water Framework Directive (2000/60/EC). Guidance document no. 8. Public Participation in relation to the Water Framework Directive. Luxemburg**:** 214.

EC (2004). "Regulation No. 911/2004 of 29 April 2004 implementing Regulation (EC) No 1760/2000 of the European Parliament and of the Council as regards eartags, passports and holding registers." *Official Journal of the European Union*. **L 163**: 65-70.

EC (2007). Factsheet: Tracing food through the production and distribution chain to identify and address risks and protect public health. Brussels.

EC (2011). "Commission Implementing Regulation (EU) No 931/2011 of 19 September 2011 on the traceability requirements set by Regulation (EC) No 178/2002 of the European Parliament and of the Council for food of animal origin." *Official Journal of the European Union*. **L 242**: 1-2.

EC (2011). Final report of an audit carried out in Germany from 12 to 23 September 2011 in order to evaluate the operation of controls over the traceability of beef and beef products.

EC (2011). "Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011." *Official Journal oftheEuropean Union*(L 304): 18–63.

EC (2015). "Regulation (EU) No 2015/262 of 17 February 2015 laying down rules pursuant to Council Directives 90/427/EEC and 2009/156/EC as regards the methods for the identification of equidae (Equine Passport Regulation)." *Official Journal of the European Union*(L 59): 51.

EPCglobal (2007). EPC Information Services (EPCIS) Version 1.0.1 Specification. . Lawrenceville, New Jersey, EPCglobal Inc.

EPCglobal (2013). GS1 Object Name Service (ONS). Version 2.0.1.

EPCglobal (2014). EPC Information Services (EPCIS) Version 1.1 Specification. GS1 Standard Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

Erl, T. (2008). *Soa: principles of service design*, Prentice Hall Upper Saddle River.

ESFRI (2016). Strategy report on research infrastructures: Roadmap 2016. . *Science and Technology*, EU.

Feng, J., Z. Fu, Z. Wang, M. Xu and X. Zhang (2013). "Development and evaluation on a RFID-based traceability system for cattle/beef quality safety in China." *Food Control*. **31**(2): 314-325.

Ferrand, N., I. Ribarova, K. Daniell, J.-E. Rougier, A. Popova, S. Vasileva and G. Abrami (2007). "Supporting a multi-levels participatory modelling process for floods and droughts comanagement." *Joumées de la Modélisation au Cemagref*: 26-27.

FI-PPP. (2013). "http://www.fi-ppp.eu/."   Retrieved June 4, 2013.

FI-Ware. (2013). "FI-Ware core platfrom generic enablers (http://catalogue.fi-ware.eu/enablers)."   Retrieved June 4, 2013.

FI-Ware. (2013). "FI-Ware Future Internet Core Platform (http://www.fi-ware.eu/)."   Retrieved June 4, 2013.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD, University of California.

FIspace. (2013). "http://www.fispace.eu."   Retrieved June 4, 2013.

Folinas, D., I. Manikas and B. Manos (2006). "Traceability data management for food chains." *British Food Journal*. **108**(8): 622-633.

Forrester, J. W. (1994). "System dynamics, systems thinking, and soft OR." *System Dynamics Review*. **10**(2-3): 245-256.

Foster, I. (2005). "Service-Oriented Science." *Science*. **308**(5723): 814-817.

Fowler, M. and J. Highsmith (2001). "The agile manifesto." *Software Development*. **9**(8): 28-35.

fTRACE. (2013). "www.ftrace.de."   Retrieved June 4, 2013.

Gandino, F., B. Montrucchio, M. Rebaudengo and E. R. Sanchez (2009). "On improving automation by integrating RFID in the traceability management of the agri-food sector." *Industrial Electronics, IEEE Transactions on*. **56**(7): 2357-2365.

Gennari, J. H., M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy and S. W. Tu (2003). "The evolution of Protégé: an environment for knowledge-based systems development." *International Journal of Human-Computer Studies*. **58**(1): 89-123.

Gilbert, N. and P. Terna (2000). "How to build and use agent-based models in social science." *Mind & Society*. **1**(1): 57-72.

Glinz, M. (2007). *On Non-Functional Requirements*. Requirements Engineering Conference, 2007. RE '07. 15th IEEE International.

Grande, E. T. G. and S. L. Vieira (2013). "Beef traceability by radio frequency identification system in the production process of a slaughterhouse." *Revista de Gestão da Tecnologia e Sistemas de Informação*. **10**(1): 99-118.

Gregersen, J. B., P. J. A. Gijsbers and S. J. P. Westen (2007). "OpenMI: Open modelling interface." *Journal of hydroinformatics(Print)*. **9**(3): 175-191.

Gregory, S. A. (1966). Design Science. *The Design Method*. S. A. Gregory. Boston, MA, Springer US**:** 323-330.

Griffioen, J., S. Vermooten, N. Kukuric, S. Vasak, S. Demuth, A. Gustard, E. Planos, F. Seatena and E. Servat (2006). "A European case-based reasoning tool to inter-compare hydrological information on water stress among drainage sub-basins." *Climate variability and change: hydrological impacts*: 91-96.

Grimm, V., U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser and D. L. DeAngelis (2006). "A standard protocol for describing individual-based and agent-based models." *Ecological Modelling*. **198**(1–2): 115-126.

Groot, S., P. J. A. Gijsbers and M. Kapetenaki (2005). HarmoniQuA - WP3 Report – second round testing**:** 125.

Gruber, T. R. (1995). "Toward principles for the design of ontologies used for knowledge sharing." *International journal of human computer studies*. **43**(5): 907-928.

GS1 (2010). Healthcare Supply Chain Traceability (White Paper).

GS1. (2013). "EANCOM <http://www.gs1.org/ecom/eancom>." Retrieved 15.04.2013.

GS1. (2013). "GLN (Global Location Number) <http://www.gs1.org/barcodes/technical/idkeys/gln>." Retrieved 15.04.2013.

GS1. (2013). "GTIN (Global Trade Item Number) <http://www.gs1.org/barcodes/technical/idkeys/gtin>." Retrieved 15.04.2013.

GS1 (2014). Core Business Vocabulary (CBV) Standard, Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

GS1 (2015). GS1 General Specifications. *Version 15 (issue 2), Jan-2015*, GS1**:** 490.

GS1 (2015). GS1 System Architecture Document: How GS1 Standards fit together, GS1 AISBL**:** 54.

GS1. (2017). "Global Standards One (GS1)." Retrieved January 25, 2017, from http://www.gs1.org/about.

GS1 EPCglobal. (2013). "Discovery Services Standard (in development) <http://www.gs1.org/gsmp/kc/epcglobal/discovery>." Retrieved 15.04.2013.

GS1 EPCglobal. (2013). "http://www.gs1.org/gsmp/kc/epcglobal." Retrieved 15.04.2013.

GS1 EPCglobal (2014). Core Business Vocabulary (CBV) Standard Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

GS1 EPCglobal (2014). EPC Information Services (EPCIS) Version 1.1 Specification. GS1 Standard Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

Gudgin, M., M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar and Y. Lafon (2007). "Simple object access protocol (SOAP) Version 1.2 Part 1: Messaging Framework (Second Edition)." *World Wide Web Consortium*.

Hammer, M. (1990). "Reengineering Work: Don't Automate, Obliterate." *Harvard Business Review*. **68**(4): 104-112.

Harrington, H. J. (1991). *Business process improvement: The breakthrough strategy for total quality, productivity, and competitiveness*, McGraw-Hill Professional.

Hartley, G. (2013). *The Use of EPC RFID Standards for Livestock and Meat Traceability*. Masters, Massey University.

Henderson, J. C. and H. Venkatraman (1993). "Strategic alignment: Leveraging information technology for transforming organizations." *IBM Systems Journal*. **32**(1): 472-484.

Hevner, A. and S. Chatterjee (2010). Design Science Research Frameworks. *Design Research in Information Systems*, Springer US. **22:** 23-31.

Hevner, A. R. (2007). "A Three Cycle View of Design Science Research." *Scandinavian Journal of Information Systems*. **19**(2): 87-92.

Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design Science in Information Systems Research." *MIS Quarterly*. **28**(1): 75-105.

Hinkelmann, K., A. Gerber, D. Karagiannis, B. Thoenssen, A. van der Merwe and R. Woitsch (2016). "A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology." *Computers in Industry*. **79**: 77-86.

Hoch, R., H. Kaindl, R. Popp and C. Zeidler (2016). *Aligning Architectures of Business and Software: Software Driven by Business Process Models and Its User Interface*. 2016 49th Hawaii International Conference on System Sciences (HICSS).

Hoffner, Y., H. Lwdwig, C. Gulcu and P. Grefen (2000). *An architecture for cross-organisational business processes*. Proceedings Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems. WECWIS 2000.

Hofmeister, C., R. Nord and D. Soni (2000). *Applied software architecture*, Addison-Wesley Professional.

Holligan, A. (2013). Horsemeat scandal: Dutch uncover large-scale meat fraud. 10 April 2013. *BBC*.

Hollingsworth, D. and U. Hampshire (1993). "Workflow management coalition the workflow reference model." *Workflow Management Coalition*: 68.

Holweg, M., S. Disney, J. Holmström and J. Småros (2005). "Supply Chain Collaboration:: Making Sense of the Strategy Continuum." *European Management Journal*. **23**(2): 170-181.

Hull, R., E. Damaggio, F. Fournier, M. Gupta, F. Heath, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya and R. Vaculin (2011). Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. *Web Services and Formal Methods: 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers*. M. Bravetti and T. Bultan. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 1-24.

Hunt, J. (2006). *Agile software construction*, Springer-Verlag New York Inc.

Inman, D., M. Blind, I. Ribarova, A. Krause, O. Roosenschoon, A. Kassahun, H. Scholten, G. Arampatzis, G. Abrami, B. McIntosh and P. Jeffrey (2011). "Perceived effectiveness of environmental decision support systems in participatory planning: Evidence from small groups of end-users." *Environmental Modelling & Software*. **26**(3): 302-309.

ISO (2004). ISO 8601: Data elements and interchange formats—information interchange—representation of dates and times. *International Organisation for Standardisation*. Geneva.

ISO. (2017). "International Organization for Standardization (ISO)." Retrieved January 25, 2017, from http://www.iso.org/.

ISO/IEC (2013). Information technology - Object Management Group Business Process Model and Notation. ISO/IEC 19510:2013.

ISO/IEC/IEEE (2011). Systems and software engineering -- Architecture description. *ISO/IEC/IEEE Standard 42010:2011*.

ISOagriNet. (2013). "ISOagriNET <http://www.isoagrinet.org/>." Retrieved 15.04.2013.

ITU-T (1999). Recommendation F.400/X.400 – series X: data networks and open system communications, message handling system and service overview.

Janssen, S., E. Andersen, I. N. Athanasiadis and M. K. van Ittersum (2009). "A database for integrated assessment of European agricultural systems." *Environmental Science & Policy*. **12**(5): 573-587.

Janssen, S., F. Ewert, H. Li, I. N. Athanasiadis, J. J. F. Wien, O. Thérond, M. J. R. Knapen, I. Bezlepkina, J. Alkan-Olsson, A. E. Rizzoli, H. Belhouchette, M. Svensson and M. K. van Ittersum (2009). "Defining assessment projects and scenarios for policy support: Use of ontology in Integrated Assessment and Modelling." *Environmental Modelling & Software*. **24**(12): 1491-1500.

Jung, K., S. Choi, B. Kulvatunyou, H. Cho and K. C. Morris (2017). "A reference activity model for smart factory design and improvement." *Production Planning & Control*. **28**(2): 108-122.

Kassahun, A., M. Blind, A. Krause and O. Roosenschoon (2008). *Integrated solution support system for water management*. iEMSs 2008, International Congress on Environmental Modelling and Software Barcelona.

Kassahun, A., R. J. M. Hartog, T. Sadowski, H. Scholten, T. Bartram, S. Wolfert and A. J. M. Beulens (2014). "Enabling chain-wide transparency in meat supply chains based on the EPCIS global standard and cloud-based services." *Computers and Electronics in Agriculture*. **109**: 179-190.

Kassahun, A., R. J. M. Hartog and B. Tekinerdogan (2016). "Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture." *Computers and Electronics in Agriculture*. **123**: 275-291.

Kassahun, A., H. Scholten and A. J. M. Beulens (2007). Collect and Share Existing Knowledge on Collaborative Multidisciplinary Scientific Research Processes. *Hands-on knowledge co-creation and sharing: practical methods and techniques*, KnowledgeBoard in collaboration with VTT - Technical Research Centre of Finland and Fraunhofer IRB Verlag**:** 347-363.

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

Kassahun, A. and B. Tekinerdogan (, submitted). "BITA*: Aligning Business Processes and IT of Multiple Collaborating Organisations based on Reference Architecture." *Business & Information Systems Engineering*.

Kidane, Y. and D. Kim (2016). *Traceability Architecture: Extending EPCIS to Enhance Track and Trace with NoSQL Data Model*. 2016 IEEE International Conference on Services Computing (SCC).

Kilger, C., B. Reuter and H. Stadtler (2015). Collaborative Planning. *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. H. Stadtler, C. Kilger and H. Meyr. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 257-277.

Kosko, B. (1986). "Fuzzy cognitive maps." *International Journal of Man-Machine Studies*. **24**(1): 65-75.

Kruchten, P. (2004). *The rational unified process: an introduction*, Addison-Wesley Professional.

Kruchten, P. B. (1995). "The 4+1 View Model of architecture." *Software, IEEE*. **12**(6): 42-50.

Kruize, J. W. (2017). *Advancement of farming by facilitating collaboration : reference architectures and models for farm software ecosystems*, Includes bibliographic references. - With summaries in English and Dutch, Wageningen University.

Kruize, J. W., J. Wolfert, H. Scholten, C. N. Verdouw, A. Kassahun and A. J. M. Beulens (2016). "A reference architecture for Farm Software Ecosystems." *Computers and Electronics in Agriculture*. **125**: 12-28.

Kunze, M., M. Weidlich and M. Weske (2011). Behavioral Similarity – A Proper Metric. *Business Process Management: 9th International Conference, BPM 2011, Clermont-Ferrand, France, August 30 - September 2, 2011. Proceedings*. S. Rinderle-Ma, F. Toumani and K. Wolf. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 166-181.

Kürschner, C., C. Condea, O. Kasten and F. Thiesse (2008). Discovery Service Design in the EPCglobal Network. *The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26-28, 2008. Proceedings*. C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern and S. E. Sarma. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 19-34.

Kywe, S. M., J. Shi, Y. Li and R. Kailash (2012). "Evaluation of Different Electronic Product Code Discovery Service Models." *Advances in Internet of Things*. **Vol.02No.02**: 10.

Laplante, P. A. (2013). *Requirements engineering for software and systems*, CRC Press.

Lattanze, A. J. (2008). *Architecting Software Intensive Systems: A Practitioners Guide*, CRC Press.

Liu, C., Q. Li and X. Zhao (2009). "Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue." *Information Systems Frontiers*. **11**(3): 201-209.

Lorenz, M., A. Zeier, H. Plattner, J. Müller and M.-P. Schapranow (2011). Discovery services in the EPC network. *Designing and Deploying RFID Applications*. C. Turcu, InTech.

Ludäscher, B., I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao and Y. Zhao (2006). "Scientific workflow management and the Kepler system." *Concurrency and Computation: Practice and Experience*. **18**(10): 1039-1065.

March, S. T. and G. F. Smith (1995). "Design and natural science research on information technology." *Decision Support Systems*. **15**(4): 251-266.

Martinez-Cruz, C., I. J. Blanco and M. A. Vila (2012). "Ontologies versus relational databases: are they so different? A comparison." *Artificial Intelligence Review*. **38**(4): 271-290.

Medini, K. and J. P. Bourey (2012). "SCOR-based enterprise architecture methodology." *International Journal of Computer Integrated Manufacturing*. **25**(7): 594-607.

Mentzer, J. T., W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith and Z. G. Zacharia (2001). "DEFINING SUPPLY CHAIN MANAGEMENT." *Journal of Business Logistics*. **22**(2): 1-25.

Meuwissen, M. P. M., A. G. J. Velthuis, H. Hogeveen and R. B. M. Huirne (2003). "Traceability and certification in meat supply chains." *Journal of Agribusiness*. **21**: 167 - 181.

Middlemis, H. (2000). Murray-Darling Basin Commission. Groundwater flow modelling guideline. South Perth. Western Australia, Aquaterra Consulting Pty Ltd.

Mitra, N. (2003). "Soap version 1.2 part 0: Primer." *W3C Recommendation*. **24**.

Moberg, D. and R. Drummond (2005). "MIME-based secure peer-to-peer business data interchange using HTTP, Applicability Statement 2 (AS2)." *The Internet Engineering Task Force (IETF)*.

Moe, T. (1998). "Perspectives on traceability in food manufacture." *Trends in Food Science & Technology*. **9**(5): 211-214.

Moltchanov, B. and O. Rodriguez Rocha (2014). *Generic enablers concept and two implementations for European Future Internet test-bed*. Computing, Management and Telecommunications (ComManTel), 2014 International Conference on.

Moore, R. V. and C. I. Tindall (2005). "An overview of the open modelling interface and environment (the OpenMI)." *Environmental Science & Policy*. **8**(3): 279-286.

Mousavi, A., M. Sarhadi, S. Fawcett, S. Bowles and M. York (2005). "Tracking and traceability solution using a novel material handling system." *Innovative Food Science & Emerging Technologies*. **6**(1): 91-105.

Muetzelfeldt, R. and J. Massheder (2003). "The Simile visual modelling environment." *European Journal of Agronomy*. **18**(3–4): 345-358.

Mukhopadhyay, T., S. Kekre and S. Kalathur (1995). "Business Value of Information Technology: A Study of Electronic Data Interchange." *MIS Quarterly*. **19**(2): 137-156.

Murphy, G. C., D. Notkin and K. J. Sullivan (2001). "Software reflexion models: bridging the gap between design and implementation." *Software Engineering, IEEE Transactions on*. **27**(4): 364-380.

Myhre, B., T. Netland and G. Vevle (2009). The footprint of food - A suggested traceability solution based on EPCIS. *RFID Systems and Technologies (RFID SysTech), 5th European Workshop***:** 1-6.

Mynetfair. (2013). "www.mynetfair.com " Retrieved June 4, 2013.

Mysiak, J., G. M. Bazzani, M. Tamaro, Y. Depietri and C. Guipponi (2011). *Decision Support for Water Framework Directive Implementation*, IWA Publishing.

Newcomer, E. and G. Lomow (2004). *Understanding SOA with Web Services*, Addison-Wesley Professional.

Nielsen, C. B., P. G. Larsen, J. Fitzgerald, J. Woodcock and J. Peleska (2015). "Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions." *ACM Comput. Surv.* **48**(2): 1-41.

Noy, N. and D. McGuinness. (2001). "Ontology Development 101: A Guide to Creating Your First Ontology." from http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

Nunamaker, J. F., M. Chen and T. D. M. Purdin (1990). "Systems Development in Information Systems Research." *Journal of Management Information Systems*. **7**(3): 89-106.

Nuseibeh, B. and S. Easterbrook (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*. Limerick, Ireland, ACM.

OASIS (2004). UDDI Version 3.0.2. **uddi_v3**.

OASIS (2006). Reference model for service oriented architecture 1.0. *OASIS Standard*. **12**.

OASIS. (2017). "Organization for the Advancement of Structured Information Standards (OASIS)." Retrieved January 7, 2017, from https://www.oasis-open.org/.

OMG (2011). Business Process Model and Notation (BPMN). **formal/2011-01-03**.

OMG (2015). OMG Unified Modeling Language (OMG UML). **formal/2015-03-01:** 752.

OMG. (2017). "Object Managment Group (OMG)." Retrieved January 25, 2017 from http://www.omg.org/.

Österle, H., J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Mertens, A. Oberweis and E. J. Sinz (2011). "Memorandum on design-oriented information systems research." **20**(1): 7-10.

Ouvans, C., M. Dumas, A. H. M. ter Hofstede and W. M. P. van der Aalst (2006). *From BPMN Process Models to BPEL Web Services*. Web Services, 2006. ICWS '06. International Conference on.

Papazoglou, M. P., P. Traverso, S. Dustdar and F. Leymann (2008). "Service-Oriented Computing: A Research Roadmap." *International Journal of Cooperative Information Systems*. **17**(02): 223-255.

Patidar, S., D. Rane and P. Jain (2012). *A Survey Paper on Cloud Computing*. Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on.

Peffers, K., T. Tuunanen, M. Rothenberger and S. Chatterjee (2008). "A Design Science Research Methodology for Information Systems Research." *J. Manage. Inf. Syst.* **24**(3): 45-77.

Peltz, C. (2003). Web Services Orchestration and Choreography. **36:** 46-52.

Piccoli, G. (2012). *Information Systems for Managers: Text and Cases*, Wiley.

Polkinghorne, R. J. and J. M. Thompson (2010). "Meat standards and grading: A world view." *Meat Science*. **86**(1): 227-235.

Premanandh, J. (2013). "Horse meat scandal – A wake-up call for regulatory authorities." *Food Control*. **34**(2): 568-569.

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, GEUS.

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., B. Nilsson, J. Brown, B. Klauer, R. Moore, T. Bech, M. Vurro, M. Blind, G. Castilla, I. Tsanis and P. Biza (2005). "Harmonised techniques and representative river basin data for assessment and use of uncertainty information in integrated water management (HarmoniRiB)." *Environmental Science & Policy*. **8**(3): 267-277.

Rezaei, R., T. K. Chiew and S. P. Lee (2014). "A review on E-business Interoperability Frameworks." *Journal of Systems and Software*. **93**: 199-216.

Ribarova, I., D. Assimacopoulos, P. Jeffrey, K. A. Daniell, D. Inman, L. S. Vamvakeridou-Lyroudia, T. Melin, P. Kalinkov, N. Ferrand and K. Tarnaki (2011). "Research-supported participatory planning for water stress mitigation." *Journal of Environmental Planning and Management*. **54**(2): 283-300.

Rizzoli, A. E. and W. J. Young (1997). "Delivering environmental decision support systems: software tools and techniques." *Environmental Modelling & Software*. **12**(2–3): 237-249.

Rosa, M. L., M. Dumas, R. Uba and R. Dijkman (2013). "Business Process Model Merging: An Approach to Business Process Consolidation." *ACM Trans. Softw. Eng. Methodol.* **22**(2): 1-42.

Rosemann, M. and J. vom Brocke (2015). The Six Core Elements of Business Process Management. *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*. J. vom Brocke and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 105-122.

Runeson, P. and M. Höst (2008). "Guidelines for conducting and reporting case study research in software engineering." *Empirical Software Engineering*. **14**(2): 131.

Runeson, P., M. Host, A. Rainer and B. Regnell (2012). *Case study research in software engineering: Guidelines and examples*, John Wiley & Sons.

Russell, N., W. M. van der Aalst and N. Mulyar (2006). "Workflow Control-Flow Patterns: A Revised View." *BPM Center Report BPM-06-22*.

Saaty, T. L. (2008). "Decision making with the analytic hierarchy process." *International Journal of Services Sciences*. **1**(1): 83-98.

Sadiq, S., G. Governatori and K. Namiri (2007). Modeling Control Objectives for Business Process Compliance. *Business Process Management: 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007. Proceedings*. G. Alonso, P. Dadam and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 149-164.

SAF. (2013). "http://www.smartagrifood.eu/." Retrieved June 4, 2013.

Sangiorgi, D. and D. Walker (2003). *The π-calculus: a Theory of Mobile Processes*, Cambridge University Press.

SCC (2012). Supply Chain Operations Reference Model (SCOR): Revision 11.0.

Scheer, A.-W. (1992). "Architecture of integrated information systems-bases for company modeling." *Berlin, 2nd edition edition*.

Schekkerman, J. (2004). *How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework*, Trafford Publishing.

Schiefer, G. (2011). "Transparency in Food: A Challenge for Research and Sector Initiatives." *International Journal on Food System Dynamics*. **2**(2): 112-113.

Scholten, B. (2007). *The road to integration: A guide to applying the ISA-95 standard in manufacturing*, Isa.

Scholten, H., A. Kassahun and J. C. Refsgaard (2004). *Structuring multidisciplinary knowledge for model-based water management: the HarmoniQuA approach*. 18th Workshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment, 6 – 8 September, 2004, Laxenburg, Austria, IIASA.

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**(5): 743-759.

Schulz, K. A. and M. E. Orlowska (2004). "Facilitating cross-organisational workflows with a workflow view approach." *Data & Knowledge Engineering*. **51**(1): 109-147.

Sein, M. K., O. Henfridsson, S. Purao, M. Rossi and R. Lindgren (2011). "Action design research." *Mis Quarterly*. **35**(4): 1099-1099.

Shanahan, C., B. Kernan, G. Ayalew, K. McDonnell, F. Butler and S. Ward (2009). "A framework for beef traceability from farm to slaughter using global standards: An Irish perspective." *Computers and Electronics in Agriculture*. **66**(1): 62-69.

Simon, H. A. (1996). *The sciences of the Artificial*. Cambridge, MA [etc.], The MIT Press.

Sjøberg, D. I. K., T. Dybå, B. C. D. Anda and J. E. Hannay (2008). Building Theories in Software Engineering. *Guide to Advanced Empirical Software Engineering*. F. Shull, J. Singer and D. I. K. Sjøberg. London, Springer London**:** 312-336.

Spanos, D.-E., P. Stavrou and N. Mitrou (2012). "Bringing relational databases into the semantic web: A survey." *Semantic Web*. **3**(2): 169-209.

Steinberger, G., M. Rothmund and H. Auernhammer (2009). "Mobile farm equipment as a data source in an agricultural service architecture." *Computers and Electronics in Agriculture*. **65**(2): 238-246.

Straub, D., M.-C. Boudreau and D. Gefen (2004). "Validation guidelines for IS positivist research." *The Communications of the Association for Information Systems*. **13**(1): 63.

Tekinerdogan, B. (2014). Software Architecture. *Computing Handbook, Third Edition*, Chapman and Hall/CRC**:** 1-16.

Thakur, M., C.-F. Sørensen, F. O. Bjørnson, E. Forås and C. R. Hurburgh (2011). "Managing food traceability information using EPCIS framework." *Journal of Food Engineering*. **103**(4): 417-433.

The Open Group (2011). *TOGAF Version 9.1*. Zaltbommel, Van Haren Publishing.

The Open Group (2013). ArchiMate 2.1 Specification (http://pubs.opengroup.org/architecture/archimate2-doc/toc.html).

The Open Group. (2017). "The Open Group." Retrieved January 25, 2017, from http://www.opengroup.org/.

Thiesse, F., C. Floerkemeier, M. Harrison, F. Michahelles and C. Roduner (2009). "Technology, standards, and real-world deployments of the EPC network." *Internet Computing, IEEE*. **13**(2): 36-43.

Triantaphyllou, E. (2000). Multi-Criteria Decision Making Methods. *Multi-criteria Decision Making Methods: A Comparative Study*. Boston, MA, Springer US**:** 5-21.

Trienekens, J. H., P. M. Wognum, A. J. M. Beulens and J. G. A. J. van der Vorst (2012). "Transparency in complex dynamic food supply chains." *Advanced Engineering Informatics*. **26**(1): 55-65.

UN/CEFACT. (2017). "UN Centre for Trade Facilitation and Electronic Commerce (UN/CEFACT)." Retrieved January 7, 2017, from http://www.unece.org/cefact.

UN/ECE (2006). UNECE Standard Bovine Meat - Carcases and Cuts. Geneva, 6–9 November 2006, UN Economic Commision for Europe. **ECE/TRADE/C/WP.7/2006/11**.

Vamvakeridou-Lyroudia, L. and D. Savic (2008). *System Dynamics Modelling: A Tool for Participatory Simulation of Complex Water Systems within AquaStress*. Proc. 4th Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs2008.

van der Aalst, W. M. (1998). "The application of Petri nets to workflow management." *Journal of circuits, systems, and computers*. **8**(01): 21-66.

van der Aalst, W. M. P., A. K. A. de Medeiros and A. J. M. M. Weijters (2006). Process Equivalence: Comparing Two Process Models Based on Observed Behavior. *Business Process Management: 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006. Proceedings*. S. Dustdar, J. L. Fiadeiro and A. P. Sheth. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 129-144.

van der Aalst, W. M. P. and A. ter Hofstede. (2011). "Workflow Patterns. http://www.workflowpatterns.com/." Retrieved December 23, 2015.

van der Aalst, W. M. P., A. H. M. Ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns." *Distributed and Parallel Databases*. **14**(1): 5-51.

van Dorp, C. A. (2004). *Reference-data modelling for tracking and tracing*, Met lit. opg. - Met samenvatting in het Engels en Nederlands
Proefschrift Wageningen, Wageningen University.

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*. Utrecht/Lelystad, STOWA/RWS-RIZA.

Verbeke, W. (2001). "Beliefs, attitude and behaviour towards fresh meat revisited after the Belgian dioxin crisis." *Food Quality and Preference*. **12**(8): 489-498.

Verdouw, C. N., A. J. M. Beulens, J. H. Trienekens and J. Wolfert (2010). "Process modelling in demand-driven supply chains: A reference model for the fruit industry." *Computers and Electronics in Agriculture*. **73**(2): 174-187.

Villa, F., I. N. Athanasiadis and A. E. Rizzoli (2009). "Modelling with knowledge: A review of emerging semantic approaches to environmental modelling." *Environmental Modelling & Software*. **24**(5): 577-587.

Voulodimos, A. S., C. Z. Patrikakis, A. B. Sideridis, V. A. Ntafis and E. M. Xylouri (2010). "A complete farm management system based on animal identification using RFID technology." *Computers and Electronics in Agriculture*. **70**(2): 380-388.

W3C (2007). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C. **REC-wsdl20-20070626**.

Weidlich, M., J. Mendling and M. Weske (2011). "Efficient Consistency Measurement Based on Behavioral Profiles of Process Models." *IEEE Transactions on Software Engineering*. **37**(3): 410-429.

Wien, J., A. Otjens and T. Van der Wal (2003). *ICT tools for participatory planning*. Food and the Environment 2003 Conference," European Federation for Information Technology in Agriculture, Debrecen, Hungary, July.

Wieringa, R., N. Maiden, N. Mead and C. Rolland (2006). "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion." *Requirements Engineering*. **11**(1): 102-107.

Wieringa, R. J., H. M. Blanken, M. M. Fokkinga and P. W. P. J. Grefen (2003). Aligning Application Architecture to the Business Context. *Advanced Information Systems Engineering: 15th International Conference, CAiSE 2003 Klagenfurt/Velden, Austria, June 16–20, 2003 Proceedings*. J. Eder and M. Missikoff. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 209-225.

Wolfert, J., C. N. Verdouw, C. M. Verloop and A. J. M. Beulens (2010). "Organizing information integration in agri-food—A method based on a service-oriented architecture and living lab approach." *Computers and Electronics in Agriculture*. **70**(2): 389-405.

Wolstencroft, K., R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi and C. Goble (2013). "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud." *Nucleic Acids Research*. **41**(W1): W557-W561.

Wong, P. Y. H. and J. Gibbons (2008). A Process Semantics for BPMN. *Formal Methods and Software Engineering: 10th International Conference on Formal Engineering Methods, ICFEM 2008, Kitakyushu-City, Japan, October 27-31, 2008. Proceedings*. S. Liu, T. Maibaum and K. Araki. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 355-374.

Yildiz, B. M., B. Tekinerdogan and S. Cetin (2012). *A Tool Framework for Deriving the Application Architecture for Global Software Development Projects*. 2012 IEEE Seventh International Conference on Global Software Engineering.

Yin, R. K. (2003). "Case Study Research: Design and Methods." *SAGE*(181): 28.

Zachman, J. A. (1987). "A framework for information systems architecture." *IBM Systems Journal*. **26**(3): 276-292.

Zachman, J. A. (2016). "The Zachman Framework for Enterprise Architecture." from http://www.zachman.com/.

Zadeh, L. A. (1994). "Fuzzy logic, neural networks and soft computing." *Communications of the ACM*. **37**(3): 77-84.

SUMMARY

When multiple organisations want to collaborate with one another they have to integrate their business processes. This requires aligning the collaborative business processes and the underlying IT (Information Technology). Realizing the required alignment is, however, not trivial and is the subject of this thesis.

We approached the issue of alignment in three steps. First, we explored business-IT alignment problems in detail in a real-life business case. This is done in order to clarify what alignment of business processes and IT systems across a collaboration network entails. Second, we provided a business-IT alignment framework called BITA* (pronounce *bita-star*). The framework provides modelling abstractions for alignment. Third, we applied the framework in two real-life case studies, including the real-life business case used in step one. By applying the framework in practice we showed that the framework can, in fact, help to address the business-IT alignment problems that we identified in the first step.

The work presented in this thesis is conducted over a number of years in the context of four large EU sponsored research projects. The projects focused on alignment problems in two very distinct application areas. Two projects were about realizing transparency systems for meat supply chains and constitute the first case study. The other two projects were about realizing multidisciplinary modelling collaboration systems and constitute the second case study. Although the projects were conducted sequentially the research questions were addressed iteratively over the years. The research methodology that shows how the framework is designed and how the case studies are applied is discussed in detail in chapter 2.

In chapter 3 we present BITA*, a Business-IT Alignment framework for multiple collaborating organisations. The main challenges in designing BITA* have been what models to consider for alignment and how to compare them in order to make explicit statements about alignment. We addressed this problem by introducing *allocation* and *alignment* modelling constructs to help the alignment process, and the concept of *business collaboration model* to represent the models that have to be aligned. We identified three groups of stakeholders for whom we designed explicit design viewpoints and associated allocation and alignment models. The Business Process to Business Process (BP2BP) alignment viewpoint is designed for business analysts who have to align diverse business collaboration process models. The IT to IT (IT2IT) alignment viewpoint is designed for software architects to align the distribution of data and IT systems across a collaboration network. The Business Process to IT (BP2IT) alignment viewpoint is designed for an interdisciplinary team of business analysts and software architects who have to align the different ways of supporting business collaboration processes with distributed IT system.

An essential element of this thesis has been elaborating how business-IT alignment problems occur in the context of multi-organisational collaboration. The case studies were used to demonstrate business-IT alignment concerns. Particularly, the details of the first case study presented in chapters 4 and 5 were used in chapter 3 to help derive the alignment framework. The case study presented an ideal problem scenario since realizing transparency across supply chains is intrinsically a collaborative effort. The second case study was used to enhance the validity of our approach. The results of the second case study are presented in chapter 6.

The alignment framework was designed during the iterative process we followed when realizing a generic transparency system for meat supply chains. To realize the required generic transparency system we needed a reference architecture. To derive the reference architecture we adapted an already existing and broadly-accepted generic reference architecture. We have to adapt the generic reference architecture in order to address specific requirements of the meat sector that were not considered in the generic reference architecture. The adaptation process made it clear that we needed models for representing business collaborations. We, therefore, introduced the notion of business collaboration model, which we used both to model reference architectures and to adapt them. Adaptation required aligning the generic reference architecture with the diverse business collaboration models adopted by the organisations that have to collaborate. The alignment framework is thus used for adapting a generic reference architecture in order to create a reference architecture that the collaborating organisations can, and are willing to, adopt.

We identified three types of business collaboration models: business collaboration process model, business collaboration IT model, and a model for representing the relationship between these two. A business collaboration process model is a business process model that spans a collaboration network. A business collaboration IT model is a model of the distribution of the IT across the collaboration network. A business collaboration process-IT model is a model of the relationships between the elements of the business collaboration processes and the elements of the distributed IT.

Each organisation is considered to adopt its own business collaboration models. For instance, different actors in meat supply chains have different views on how chain-wide transparency should be realized. Which business processes and IT systems each organisation has to deploy and use depends on the business collaboration models each food operator adopts. If two different food operators adopt the same set of business collaboration models, they are aligned; otherwise they are misaligned. Hence, alignment entails comparing the different business collaboration models adopted by the participating organisations. The results of the alignment process are explicit statements about how convergent or divergent the organisations are from the chosen generic reference architecture. The explicit statements of alignment guide how best the generic and the corresponding organisational business collaboration models can be adapted to create a better state of alignment.

To further enhance the validity of the overall approach the second case study was conducted. The second case study was a retrospective investigation of two past research projects focusing on aligning environmental modelling processes and IT systems. A retrospective case study was chosen because launching a new business-IT alignment project involving multiple collaborating organisations was not feasible. The projects were undertaken to support the European Water Framework Directive, which mandated, among other things, participatory, multidisciplinary, river-basin wide and model-based studies to manage the water resources of Europe. The directive particularly required a collaborative approach to building environmental decision support systems and to deriving methodologies for applying existing decision support systems. We applied BITA* to aligning environmental modelling processes and IT systems in order to evaluate the suitability of the framework to addressing alignment problems in other application areas.

The contributions of the thesis are summarized in chapter 7. The contributions include a number of design artefacts, which can be grouped into four categories: constructs, models, methods, and instantiations. The contribution in the first category includes the conceptualization of allocation and alignment. The contributions in the second category include allocation and alignment models, and reference architectures. Allocation models are representations of business collaboration models in a form that can be compared and are the basis for alignment modelling. The main contribution in the third category is the BITA* systematic approach to alignment modelling. The contributions in the fourth category are the software systems developed with the help of the reference architectures.

ACKNOWLEDGEMENTS

At last, the PhD project has come to a successful completion. The PhD journey started a decade ago on a different topic and come to an end with this thesis along a winding road. Many people have helped me and I wish to express my gratitude. First of all, I would like to thank my promotors Adrie Beulens and Bedir Tekinerdogan. Adrie, I would like to thank you for your endless patience and support. You acquired the SAF research project and you assigned me to the project, an assignment that made this PhD project possible. You were willing to review the last versions of this thesis even after an eye operation; I greatly appreciate that. Bedir, you are a very effective and resourceful supervisor. You started to supervise me in 2015 and in less than two years I was able to complete the thesis. The remarkable progress would not have been possible without your commitment. I feel very fortunate to have you as my promotor.

I would to thank my long time colleagues Huub Scholten and Rob Hartog with whom I collaborated in a number of research projects since 2002. Rob, the impetus for this thesis is our paper that was published in 2014 and without that fresh start this thesis wouldn’t have been possible. We wrote two of the papers that are now part of this thesis and you carefully and critically reviewed the entire thesis. You are a very good supervisor and I cannot express enough how grateful I am for your help. Huub, it is because of you that I joined Wageningen University. I would like to thank you for the trust you had in me. We had a very intensive collaboration during the HarmoniQuA and AquaStress projects. I travelled to many different countries and come to know and work with many researchers from all over Europe because of you. I learned a lot from your courageous and optimistic approach to problems and great organisational skills.

The PhD journey started formally in 2006 thanks to Gert Jan Hofstede. You told me friendly and politely—almost as if I will be doing you a favour—in the canteen of Leeuwenborch just after we had a lunch that I can, if I wish, attend the PhD introductory course you were about to start to teach. Because of you I formally became a PhD candidate. Our collaboration on the QChain project has enabled me shift from environmental modelling to supply chain management research domain. You encouraged me to collaborate with Bedir in 2012; a collaboration that became possible in 2015. You are very wise and I feel fortunate to have you as a colleague.

The work on this PhD was greatly facilitated by my appointment as an independent researcher in 2013 by Jack van der Vorst, now director of the social sciences department, then interim chair of INF. Jack, I am very grateful for the promotion which made this thesis possible. I would like to thank you also for entrusting me with full responsibility in KIGO and SCALE projects. You have been up front and direct in your supervisions and feedbacks which helped me to improve myself.

I would like to thank my INF colleagues who have been great support to me in various, and often subtle, ways; I am very grateful for that. I owe much of my teaching skills, which hasn't come naturally to me, from collaboration with Sjoukje Osinga and Mark Kramer. You also back me up when I have to leave for project meetings, conferences or family visits. I also got a lot of support from Marten Zijp who is the coordinator of most of our courses and thesis projects. Maarten, I feel I can always ask you for any information regarding educational rules and procedures. Sjoukje, Mark and Maarten, you enabled me to focus on this PhD project for which I am very grateful. I also would like to thank Gerard Moerland, Jacques Wolfert, Cor Verdouw and Jan Willen Kruize for the research collaboration in various occasions.

I would like to thank all current and former secretaries and deputy administrators, particularly Natasja Ariesen, Ilona van Veen and Leonie van Scherrenburg, who have always been supporting me with all administrative tasks. You also neatly archived the required documents, including the letter of the Dean of Sciences, which I had lost. I would like to thank all former and current researchers and colleagues of INF and ORL. You have been or are still part of my daily office experience and some of you have collaborated with me in teaching and research. You have been supportive in many different ways which I cannot reduce to one single attribution. And, it is lovely to see and feel your genuine expression of pleasure at the completion of this thesis.

This thesis is a result of large scale collaboration. I wish to extend my gratitude to all researchers who collaborated with me in FIspace, Smart Agri Food, AquaStress and HarmoniQuA projects. Particularly, I wish to thank members of GS1 and European EPC Competence Center (EECC) from Germany who have been very kind in explaining their work on transparency during several meetings and providing me with access to their transparency software systems.

I would like to thank all individuals who have taken some time to review the manuscript or help me in any way they could. I would like to thank Sjoukje, Natasja and Gert Jan for reviewing parts of the thesis at one time or another. I would like to thank Baukje Kruize for working on the outline of the thesis. I wish to thank Jacob Spoelstra, a fellow member of the Works Council of SSG, for organising a visit to a ‘model’ dairy farm at Stiens in Friesland, Jeroen Reijers of Vion Food Group for inviting me to visit the slaughter facility in Boxtel, and Albert Winkel of Nutrico for organising a visit to an animal feed premix factory in Putten. I gained a good insight into food processing from these extended visits.

Any hard work will not be possible without the support of family and friends. I would like to thank Andrea and Ruben. You have for so long been very accommodating and you hold me to higher standards. I wish to thank my parents who raised me through some tough times and (probably as a result of that) raised me to be tenacious. I would like to thank all my friends. I feel lucky that I belong to such a multicultural environment. I am so grateful to all of you (I wish I could list all of you but avoided to do so for fear of missing names) for your company and friendship which made this thesis possible.

Ayalew Kassahun — Wageningen, May 2017

About the Author

Ayalew Kassahun was born in Dessie, Ethiopia on July 1st, 1969. He holds a BSc in Civil Engineering from Addis Ababa University, MSc in Water Management from Dar es Selam University and another MSc in Soil Physics and Agrohydrology from Wageningen University. He is currently employed as lecturer and researcher at the Information Technology Group of Wageningen University. He has previously worked for Baan Company (now Infor), Enterprise Resources Planning (ERP) suite developer, for three years; and InfoRay, Business Intelligence (BI) suite developer, for almost two years. At Wageningen University he teaches software engineering, programming in Python and applied information technology. Ayalew Kassahun has been involved in a number of EU sponsored research projects which, directly or indirectly, contributed to the thesis. The projects and their web addresses are: HarmoniQuA (harmoniqua.wur.nl), AquaStress (www.aquastress.net), SAF (www.smartagrifood.eu/), SCALE (www.sfcplatform.eu), FIspace (www.fispace.eu) and IoF2020 (www.iof2020.eu). The author's scientific publications can be found by searching for Ayalew Kassahun at Google Scholar (scholar.google.com)

Ayalew Kassahun

Wageningen School of Social Sciences (WASS)

Completed Training and Supervision Plan

Name of the learning activity	Department/Institute	Year	ECTS *
A) Project related competences			
Communication in Interdisciplinary Research	WGS	2008	1.1
Presentation Skills	Language centre, WUR	2007	1.5
WebSphere Application Server - Programming	Getronics	2003	1
Microsoft: MS2609: Introduction to C# Programming,	Master-IT	2005	1
Microsoft: MS2124: .NET Framework 1.1	Master-IT	2005	1
Cor web services for Java EE developers	TwiceIT	2007	1
OpenMI – Open modelling interface	WL Delft Hydraulics	2006	1
Writing project proposal	MGS	2007	6
B) General research related competences			
Mansholt Introduction Course	WU	2006	1.5
Conference presentations			
'Support for model based water management with the HarmoniQuA Toolbox'	iEMSs	2004	1
'A knowledge base system for multidisciplinary model-based water management'	iEMSs	2006	1
'Support for collaborative multidisciplinary research processes: a knowledge base approach'	CSM, Austria	2007	1
'Integrated Solution Support System for Water Management'	iEMSs	2008	1
'Chaining Environmental Web Services: Mapping Problem Solving Tasks to Web Services'	CSM. Austria	2009	1
'Towards a service-oriented e-infrastructure for multidisciplinary environmental research'	iEMSs	2010	1
'Transparency in meat supply chains: An information system for tracking, tracing and awareness of meat (TTAM)'	Igls Forum	2013	1
'Measuring Sustainability Performance of Food Production and Logistics'	WICaNeM, Italy	2014	1

'Realizing chain-wide transparency in meat supply chains using web applications based on GS1 standards'	EFITA/WCCA/CIGR, Poland	2015	1
'Architecture Viewpoint for Modeling Business Collaboration Concerns using Workflow Patterns'	ICSOFT-EA, Portugal	2016	1
'Adopting Workflow Patterns for Modelling the Allocation of Data in Multi-Organizational Collaborations'	DATA 2016, Portugal	2016	1
C) Career related competences/personal development			
Designing Group Assignments	WUR	2015	0.5
Voice Matters: Voice and Presentation Skills	WGS	2007	0.4
Course Lecturing	WUR	2008	1
Presentation and demonstration to stakeholders of case studies	Divers	2004-2016	2
Organising knowledge dissemination programme to promote green and socially responsible chains	WUR/KIGO-MVK	2013	2
Teaching			
- Software Engineering (INF-32306)	WUR	2011 -	1
- Programming in Python (INF-22306)	WUR	2011 -	1
- Applied Information Technology (INF-20806)	WUR	2013 -	1
- Big Data	WUR	2016-	1
Total			**36**

*One credit according to ECTS is on average equivalent to 28 hours of study load

Colophon

The research leading to this thesis received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) within the FI-PPP’s project SAF under grant agreement n° 285 326 and project FIspace under grant agreement n° 604 123, and Horizon 2020 Programme (H2020/2014-2020) within the project IoF2020 under grant agreement No. 731 884.

Cover Design by:
Remco Wetzels

Printed by:
Proefschriftmaken.nl || Digiforce Vianen