Improving support for greenhouse climate management: an exploration of a knowledge-based system

Promotoren:
dr. ir. H. Challa
hoogleraar Tuinbouwbedrijfstechnologie
ir. M.S. Elzas
emeritus hoogleraar Informatica in het bijzonder Kennissystemen

Co-promotor:
dr. R. Martin Clouaire
directeur de recherche, Unité de Biométrie et Intelligence Artificielle, INRA, France

Stellingen

1. Het beschouwen van de ondersteuning van de klimaatbesturing uit het oogpunt van de tuinder is een paradigmaverandering in het huidige tuinbouwkundig onderzoek.
   Deel 1, dit proefschrift

2. Het grote aantal parameters dat de tuinder bij de huidige generatie kasklimaatcomputers kan instellen, is geen bevestiging voor het weinig geavanceerd zijn van het gebruikte besturingsalgoritme (Van Henten 1994, stelling 6), maar een gevolg van de (gewenste) flexibiliteit van deze systemen.
   Hoofdstuk 2, dit proefschrift

3. De impliciete vooronderstelling (bijv. Martin Clouaire *et al.*, 1993; Seginer, 1993; Tap *et al.*, 1996) dat een optimaliserend systeem betere setpoints of sturingen kan bepalen dan de tuinder met zijn huidige systeem weet te verzinnen, is vooralsnog niet aangetoond.
   Hoofdstuk 4, dit proefschrift

4. Constraint reasoning is vooralsnog de enige techniek die voldoende krachtig is om de kwalitatieve en kwantitatieve kennis uit het probleemdomein geïntegreerd binnen een besturingsconcept voor de kasklimaatbesturing toe te passen.
   Deel 2, dit proefschrift

5. Technologische vernieuwing komt pas na overtuiging van de potentiële gebruikers; het is zodoende evenzeer een sociaal als een technisch proces.

6. De enige validatie van een besturingsconcept is de toepassing ervan in de dagelijkse praktijk.

7. Voor een calculerende tuinder is energiebesparing geen issue.

8. Statistieken van de financiering van wetenschappelijk onderzoek veronachtzamen de aanzienlijke bijdrage van de sociale uitkeringsinstanties.

9. De kosten die gemaakt worden voor de ontwikkeling en het gebruik van "gewasgroeimodellen" in het wetenschappelijk onderzoek staan in geen verhouding tot de kosten die gemaakt worden voor het toepassingsrijp maken van deze modellen voor de tuinbouwpraktijk.

10. Nuttige momentane 'klimaatbegrenzingen' bestaan niet.

11. Ex ante keuze voor een modelleerparadigma bemoeilijkt de kijk op een probleem en vormt een risico voor een succesvolle aanpak ervan.

12. Alleen planningssystemen die toelaten dat de slimheid van de menselijke planner nuttig kan worden toegepast, zijn werkelijk succesvol.

13. Far the better an approximate answer to the *right* question, which is often vague, than an exact answer to the *wrong* question, which can always be made precise.
    John W. Tukey, 1962. Ann. Math. Stat., vol.33, p.13.

Stellingen behorende bij het proefschrift:

"IMPROVING SUPPORT FOR GREENHOUSE CLIMATE MANAGEMENT: AN EXPLORATION OF A KNOWLEDGE-BASED SYSTEM"

Peter J. Schotman

Wageningen, 27 maart 2000

P.J. Schotman

Improving support for greenhouse climate management: an exploration of a knowledge-based system

Proefschrift
ter verkrijging van de graad van doctor
op gezag van de rector magnificus van
Wageningen Universiteit, dr. C.M. Karssen,
in het openbaar te verdedigen
op maandag 27 maart 2000
des namiddags te vier uur in de Aula.

Het onderzoek dat aan dit proefschrift ten grondslag ligt, is mede mogelijk gemaakt door:

de NOVEM "Nederlandse Onderneming Voor Energie en Milieu b.v."
het Ministerie van Landbouw, Natuurbeheer en Visserij

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Schotman, P.J.

Improving support for greenhouse climate management: an exploration of a knowledge-based system / P.J. Schotman - [S.1. : s.n.]. Fig., Tab.

Thesis Wageningen - With ref. -With summary in Dutch.

ISBN: 90-5808-214-8

Subject headings: greenhouse climate control, decision support, tomato production

ABSTRACT

Schotman, P.J. Improving support for greenhouse climate management: an exploration of a knowledge-based system. Dissertation Wageningen University, Wageningen, The Netherlands. 248 pp. Including a Dutch summary.

This thesis discusses automated support for operational management of greenhouse crops and proposes a knowledge-based system to support the grower in his operational management task.

Operational management is defined as the day-to-day decision making processes which directly or indirectly lead to activities that influence the growth and development of the crop. To improve automated support for operational management, the growers operational decision making has been analyzed in the light of theory related to problem solving. The analysis of the task environment has resulted in a model of the decision processes within operational crop production management. This model has been based on the intelligence - design - choice cycle of Simon (1997). During the design and choice phases of this model the grower has to convert his observations at the crop and environment level into actions that can be implemented at the control level. Since this conversion is considered a complex and knowledge intensive task, a knowledge-based system is proposed to support the grower. The main idea behind the approach is to allow the grower to tell the system what objectives it must pursue and have the system deduce the required device settings at the control level. As these objectives may be situated at the crop, environment and control level, both domain knowledge as well as a suitable inference mechanism is required to realize such an approach.

Analysis of the knowledge in the domain of crop production shows that this knowledge is, or can be made available. Regarding the latter, the Blossom-end Rot example shows that knowledge can be made available in a suitable format.

With respect to the inference mechanism past approaches have been surveyed. Based on the results of this survey, the characteristics of the inference problem, and the attributes of the domain knowledge, it has been concluded that constraint reasoning fits the requirements best.

Simulation experiments with a prototype implementation show that the constraint reasoning can indeed be used as inference mechanism, however it is argued that the amount of work needed to realize and implementation in practice is formidable.

Keywords: climate control, tomato production, constraint satisfaction, interval constraint reasoning, operational management, *Lycopersicon esculentum* Mill., horticulture, crop growth models, Blossom-end rot, BER.

DANKWOORD

Nu dit proefschrift afgerond is, wil ik diegene bedanken die hier direct of indirect een bijdrage aan heeft geleverd.

Allereerst wil ik iedereen bedanken voor het opgebrachte geduld en het regelmatig vragen naar de status van 'mijn boekje'. Deze constante aandacht, hoewel niet altijd even plezierig, heeft er zeker toe bijgedragen dat dit werk in ieder geval in hetzelfde millennium is afgerond, als het is aangevangen.

Persoonlijk heb ik mijn onderzoek als een constante zoektocht ervaren, waarbij goede navigatie onmisbaar is om een weg te vinden in de nog niet in kaart gebrachte delen van het onderzoeksterrein. In het bijzonder wil ik mijn promotoren Hugo Challa en Maurice Elzas bedanken voor de geleverde navigatie tijdens deze zoektocht.

I owe much gratitude to Roger Martin-Clouaire for his constant support during my research. I especially remember the numerous discussions during my three months stay at your lab in Toulouse. These discussions were a real source of inspiration and motivation for me. At the same time, I would also like to thank you and your colleagues of the Unité de Biométrie et Intelligence Artificielle (INRA) for the warm hospitality I experienced. Furthermore, I would like to acknowledge Marc Tchamitchian for our pleasant cooperation while writing the survey.

Klaas Jan Leutscher en Rob Hartog wil ik bedanken voor de - vooral mentale - ondersteuning tijdens de beginfase van mijn onderzoek. In dit verband wil ik ook het maandelijkse Overleg Modellen Onderzoek van de (voormalige) vakgroep Tuinbouwplantenteelt niet onvermeld laten.

Huub Scholten en Alexander Udink ten Cate wil ik bedanken voor de aangename samenwerking tijdens onze poging het neusrotprobleem bij tomaat met behulp van SENECA te modelleren. Ep Heuvelink wil ik in dit kader bedanken voor zijn kritische blik op de modelrelaties.

De heren L. Blok, W. van Schie, en R.. van Amersfoort waren onmisbaar voor mijn inzicht in de teelt van tomaten en de werkwijze van tuinders; ik wil hun bedanken voor het feit dat ik van hun inhoudelijke expertise gebruik kon maken. Daarnaast wil ik Ad de Koning bedanken zijn suggesties met betrekking tot hoofdstuk twee.

De firma's Bolesian BV en LCN Planning & Scheduling BV ben ik dankbaar voor de vrijheid om ook tijdens kantooruren - weliswaar onbezoldigd - aan de afronding van dit proefschrift te werken.

Last but not least wil ik Caroline en mijn ouders bedanken voor hun vertrouwen dat het wel goed zou komen met dit boekje...

TABLE OF CONTENT

1. GENERAL INTRODUCTION

In this chapter the reader is introduced in the subject of this thesis, the objectives and research questions are explained, and the content of the thesis is outlined.

1.1 Greenhouse crop production

The cultivation of horticultural crops in a protected environment like a greenhouse is an important economic activity in the Netherlands. In 1992, approximately 13,000 horticultural firms produced a large variety of horticultural products on approximately 10,000 hectares. The value of the products produced by these firms was approximately 7 billion Dutch guilders (Min. LNV, 1997).

The tomato has traditionally been among the largest horticultural crops. In 1992 it covered approx. 1500 ha, although its area has decreased since then, it still is one of the major Dutch crops. In this thesis, the tomato crop acts as the example crop to refer to, not only because it was the largest crop when this research started, but also it because its production characteristics and cultivation practices are in many aspects similar to other important vegetable crops (sweet pepper and cucumbers).

1.2 Improving operational crop production management and its support

Greenhouse crop production originated from more general forms of protected cultivation. The earliest forms of protected cultivation started well before the beginning of the Christian era with the recognition that crops produced better when they where sheltered from harsh external conditions. For instance, during the Roman era cucumbers were grown in mobile baskets that could be brought into the sun and withdrawn into transparent, mica shelters during wintry days (Stanghellini, 1987). Today, like in the Roman days, modern greenhouse cultivation also requires taking appropriate actions to profit most from the opportunities which the external - light - conditions offer. Given the high production levels needed to cover the fixed and variable costs, determining what actions to take is all the more crucial in contemporary tomato production.

The determination of appropriate actions can be seen as the core of operational crop production management. Operational crop production management comprises the day-to-day decision making activities of the grower, in which the actions that result from the decisions taken by the grower, directly or indirectly influence crop growth and development. The following areas are part of operational crop production management (section 2.2.4):

- management of the shoot environment of the crop (climate management),

- management of the root environment of the crop (irrigation and nutrition management),
- management of pest and diseases, and
- management of manual operations (*e.g.* harvesting, leaf pruning)

Although it will be argued later that an integrated view of these management areas is essential for proper computerized support, in this thesis focus will be placed on climate management.

At modern horticultural firms the grower uses a climate control system to help him carry out his climate management task. The principal task of this system is to regulate the environment according to the instructions entered by the grower. Although these systems are continuously being improved, their fundamental underpinnings are more or less the same, and are based on approximately 25 year old technology (Bontsema, 1995; Van Meurs, 1995). Their main characteristics can be summarized as follows:

- The systems are general purpose control systems for the greenhouse environment and do not incorporate crop-specific knowledge or models.
- A typical climate control system, which uses a form of PID control, can still be seen as an operator dealing with control devices using heuristic-based control strategies.
- The number of variables the grower can set in his systems have steadily increased. At present there are well over a hundred of variables that could be set per greenhouse compartment, owing to the interrelations between these variables, managing them carefully is increasingly difficult.

Above characteristics are increasingly seen as weaknesses, and have, in conjunction with the advances in control engineering, decision support and in fields related to horticulture, encouraged many researchers to investigate possibilities for improvement (see *e.g.* Udink ten Cate *et al.*,1978 for an early study).

Apart from above system-related opportunity for improvement, there is also an operational one. This opportunity stems from the energy intensive nature of greenhouse crop production. It has been observed in practice that quite a number of growers use control settings (like a fixed minimum pipe temperature), whose usefulness have been questioned, because the objectives that these settings serve are not always clear, or even seem inappropriate at times (*e.g.* Bakker, 1991). Furthermore, it has been reported in professional literature that the relation between energy use and yield varies considerably among growers, *i.e.* some growers use much less energy to produce the same amount of tomatoes per m^2. These observations suggest that climate management could be improved, possibly through better computerized support.

Finally, there is the strategic goal to save our (limited) energy resources through better climate management, this goal motivated much research, especially in the mid-eighties when energy prices were particularly high. At this moment, while energy prices are relatively low there is also an incentive from the society to develop more sustainable production systems. The large contribution of the greenhouse industry in the Netherlands to the national energy consumption, against the background of international concern about the global warning, has forced Dutch growers to consider this problem and face their responsibility.

1.3 Thesis objective

From the above it may be concluded that there are ample reasons for improving climate management. Investigating how climate management could be improved by means of computerized support, is the subject of this thesis. This objective is developed in a bipartite manner.

First, the principles of climate management, and methods that can be used to improve climate management will be analyzed. This involves: *i*) investigating the task environment in which improvements of climate management must be embedded, *ii*) investigating the available knowledge that can be used to improve climate management, and, *iii*) investigating methods and techniques that have been suggested in the past for improving climate management.

Second, on the basis of this analysis, a concept for a knowledge-based computerized support system will be proposed. Within this concept the opportunities and practicability of modern control engineering and decision support techniques should be carefully balanced, that is, the application conditions of the chosen approach should be realistic and reachable in the nearby future. These application conditions refer both to the intrinsic demands of the chosen approach, as well as to the task environment in which improvements of climate management must be embedded.

The main difference between this work and other works reported in literature (*e.g.* Martin-Clouaire *et al.*, 1993a,1993b; Van Henten, 1994; Van Straten and Challa, 1995; Seginer, 1996) is the focus on the analysis of the task environment and implications of the task environment on the structure and behavior of a new computerized support system, while other approaches focused more on techniques and did not consider the task environment in its entirety.

Related to our focus on the task environment, climate management is approached as an integral part of operational crop production management and the consequences of the interrelations between the management areas mentioned in the previous section are considered.

1.4 Research questions

The objective explained in the previous section has resulted in the following research questions:

1. *What is the managerial context in which computerized support must be embedded?* Computerized support is a means, not a goal in itself, thus whenever computerized support is suggested, a clear notion of the tasks to be supported,

must be available. Without sufficient insight in the objectives and working methods of the grower, a suitable support tool is difficult - if not impossible - to construct. Therefore, in chapter two, the tasks of the grower within operational crop production management are analyzed from a 'support' perspective, that is, how they may be supported effectively.

2. *What kind of knowledge is available for knowledge-based computerized support?*
   Knowledge about greenhouse crop production (and more specifically tomato production) is diverse in nature. Moreover, because crops are not man-made, we have only limited insight in the processes that take place in the crop. Since, various types of computerized support require its embedded knowledge to conform to specific requirements, a clear insight into the attributes of the knowledge available in the domain of greenhouse crop production is necessary. Such insight facilitates the choice of an appropriate computerized support approach.
3. *What should a support system look like and how should it behave in order to offer valuable knowledge-based computerized support?*
   Research is not an isolated activity, therefore, before embarking on a new scientific journey one should have a clear notion of the ideas that have been proposed and pursued in the past. A review of the applicable literature shows that our field of study has been particularly active in the recent past. Insight in past approaches allows one to learn from them and prevents one from pursuing potentially interesting but less fruitful ones.
4. *What is the relevance of new modelling and problem solving techniques for computerized support in the management task, and how can they be embedded in an overall system design?*
   The investigations of the first three research questions form the proper foundation for this final question. Here a concept of a computerized support system will be proposed that is in agreement with above findings.

1.5 Research strategy

1.5.1 The managerial context into which computerized support must be embedded

To determine the application context in which a next generation support system must work, the objectives and operational tasks of the grower must be precisely known. To acquire insight into the objectives and working practices of the grower, interviews and discussions with growers, consultants of extension service agencies and other specialists in the field of tomato production must be carried out. Afterwards the analyzed results are checked by experts in the field and placed alongside the few other sources that have investigated the working practices of the grower. Furthermore, the objectives and tasks of the grower are placed in the context of literature on decision support. This activity results in a model that depicts the activities of the grower within operational crop production management. Finally, this model is used to identify the tasks of the grower for which additional computerized support could be beneficial.

1.5.2 Domain knowledge in greenhouse crop production

For knowledge-based computerized support one needs knowledge that conforms to the specific demands of the inference technique(s) chosen.

Knowledge is remarkably diverse, it ranges from informal and qualitative procedural statements to numerical equations that specify cause and effect relations. To have a clear notion of what kind of knowledge is available, or can - without too much effort - be acquired, a literature survey is carried out. This survey aims at acquiring detailed insight in the structural and relational properties of the body of knowledge available in domains related to greenhouse crop production.

1.5.3 Previous system proposals

To be able to take a step forward it is essential to acquire sufficient insight in the strong and weak elements in earlier attempts in this field. To acquire this insight, a literature survey is carried out on this subject as well.

1.5.4 Synthesis, formulating a concept for computerized support

Formulating a concept for a new software system requires synthesizing from the insights obtained during the analysis phase. These insights are usually formulated as requirements which the new system should satisfy. Once the requirements are clear, a global functional design can be proposed. In this design the tasks of the system are logically grouped and divided over functional modules.

As so often in research related to software engineering, designing and prototyping a computerized support system involves a mixture of a few scientifically innovative and many practical considerations. Given the nature of this work, our main attention is concentrated on the scientifically most interesting issues in the development of a system concept. These issues are: *i*) the choice of an appropriate inference technique; *ii*) the description of the chosen technique in the light of its application conditions (*i.e.* modelling the problem of climate management in the 'language' of the technique at hand), and, *iii)* the analysis of the behavior of the chosen technique(s) in the light of the requirements. Investigation of the behavior of the inference technique(s) is carried out by means of simulation.

1.6 Reading suggestions

This thesis contains two main parts: an analysis part and a design part. Each part is concluded with an epilogue that compiles its main results. The overall structure of this thesis follows the order in which the research strategy has been stated. The thesis ends with a general discussion in which the research objective will be reviewed and future research directions will be discussed.

Although the thesis is written to be read from cover to cover, various 'short cuts' are possible. Three of them are discussed here. First, the reader who is mainly interested in grower-oriented issues and the practical context of future computerized support in greenhouse crop production should focus on chapters two, five and six. Second, the reader who is more interested in an application of constraint reasoning (as a specific form of optimization) may read chapter six to acquire sufficient insight in the context in which constraint reasoning is applied, continue in chapter seven for a detailed description of the technique and the implementation of the prototype, and proceed with chapter eight to see its behavior. Third, the reader who is mainly interested in the state of affairs regarding the available domain knowledge for computerized support should focus on chapter three and appendix one.

PART I

PROBLEM ANALYSIS

2. GREENHOUSE CROP PRODUCTION IN A MANAGEMENT CONTEXT

2.1 Introduction

In this chapter crop production will be defined and analyzed in a management context. Above all managing crop production implies managing the processes that take place in the crop by creating proper environmental conditions and carrying out crop handling activities such that these processes proceed in a desired direction.

Managing the crop's shoot environment is considered the most important and most difficult aspect of operational crop production management. In later chapters, this part of the grower's management task will therefore be emphasized upon. Managing the crop's shoot environment cannot be seen separately from other means to influence crop production. Consequently, in this chapter, all activities influencing crop production and their interactions will be taken into account. The analysis will be oriented toward the question of *how* a grower's operational management task can best be supported.

Managing the crop's root and shoot environment has been supported by control systems since the early seventies. These systems offer the grower significant advantages over manual operation because: *i*) they save time, the grower can implement a control strategy (a combination of setpoints for the various variables over time) and have the control system execute it autonomously; *ii*) complex - multivariate - strategies, providing a fast response to changes in the outside conditions, can be exploited; *iii*) control actions can be applied more efficiently in terms of energy expenditure (Van Mansom and Rieswijk, 1995).

The principal idea behind our work is that the present-day support in operational crop production management can be extended. However, before deciding about how computerized support for greenhouse growers may be organized, a detailed understanding of the decision making activities at a horticultural firm is considered necessary. This is in close agreement with the ideas of Keen and Scott Morton (1978) who state that to be able to improve a decision process, one must first define and analyze it. The objective in this chapter is therefore to develop a descriptive framework of the grower's management problem. In later chapters, this framework will be used to develop a design for a computerized support system.

A support system designed for individual growers on different horticultural firms must be flexible enough to allow for the multiple management styles that have been observed in practice (Spaan and Van der Ploeg, 1992). Furthermore, the design should be general enough to be applicable in other (vegetable) crops, since implementing such a system and putting it to practice will not be commercially viable when its applicability is very limited.

The objective of this chapter will be worked out in the following way: first, an overview of tomato production will be given. Second, the boundaries of the management problem that plays a central role in this thesis will be stated. Third, important characteristics of operational crop production management will be discussed, among others: *i*) the way growers observe their crop; *ii*) the growers operational objectives; and *iii*) the growers problem solving and decision making behavior, will be analyzed. Finally, based on literature regarding problem solving, a model of operational crop production management will be proposed. This model will be used as a reference in the subsequent chapters.

2.2 Tomato production

In order to familiarize the reader with tomato production an outline of operational crop production management will be presented. In subsequent sections, the decision making processes of the grower will be described in detail.

2.2.1 The crop compared to a man-made production facility

Taking a functional point of view, greenhouse crop production can be compared with production processes that take place in fully man-made production facilities.

Some parts of crop production resemble a job-shop production facility, where sequences of activities can be observed, in which each activity has its own theoretical maximum production rate. Between these activities there are stocks of intermediary products, that potentially allow each individual activity to continue smoothly, that is, an activity should not have to wait on its predecessor because its inputs are readily available. In this analogy the activities resemble the processes taking place in a plant, while intermediary stocks can be compared to the buffers between processes. For instance, the uptake and transport of nutrients can be seen in this way.

Some parts of crop production better resemble an assembly line production facility. Here activities are carried out on a product that has been placed on a conveyor belt. The quality of the final product depends upon the quality of the individual activities along the assembly line; the overall performance with respect to quantity depends upon the speed of the conveyor belt. The growth of individual fruits could be viewed in this way. In this analogy, it is the developmental rate of the fruits that could be compared to the velocity of the assembly line, while their size and quality are determined by processes that take place during the course of development.

Another resemblance between crop production and production in fully man-made production facilities is the presence of feedback mechanisms. Like in factories, crops adjust their process rates in a feedback-like manner, for instance, in case of tomato or cucumber production, flower and fruit abortion seems to be directly related to the (inadequate) supply of assimilates (De Koning, 1994; Marcelis, 1994).

Differences between crop production and production processes in fully man-made production facilities can also be observed. For instance:

- since crops are not man-made we know much less about the individual processes as compared to the individual activities in a factory,
- as opposed to the activities in a factory, the processes in a crop cannot usually be controlled separately and some cannot be controlled at all (*e.g.* crop aging), and
- the uncertainty and significance of external disturbances (weather) in crop production is large. The influence of these external factors (and their associated uncertainty) is generally much larger than the influence of external factors in man-made production facilities.

2.2.2 Characteristics of greenhouse production

Early in history man discovered the benefits of protecting crops from unfavorable outdoor conditions. One of the first examples of man-made protection in commercial production were (brick) walls, to protect crops against the influence of wind. Much later, Dutch Lights (or flat glass) were invented, which were improved and gradually developed to contemporary Venlo-type glasshouses. Today, we see a large variety of more or less sophisticated greenhouses in which all kinds of horticultural crops are grown. In the Netherlands, as opposed to warmer regions, glass is the primary greenhouse cover. Although one may easily think that all glasshouses are more or less equal, one should realize that for management and control practices no greenhouse is alike.

With respect to their influence on management and control practices, important differences between the production conditions in greenhouses are caused by: *i*) (geographical) location; *ii*) orientation of the greenhouse; *iii*) the air exchange and light-transmission characteristics of the greenhouse; *iv*) the height of the greenhouse; *v*) production or growing system; and *vi*) the available control equipment[1]. Small differences in these characteristics may influence growers' practices significantly.

2.2.2.1 The shoot environment

The shoot environment is the above ground (growing medium) environment that can be characterized by its ability to vary relatively rapidly. With respect to the crop's shoot environment, growers can control the four major[2] climate variables:

---

[1] The control equipment is defined as the set of available control devices or actuators. The control system is the (computerised) interface between the grower and his control equipment.

[2] NO_X-concentration, CO-concentration could be considered minor climate variables.

light intensity (I_0 in W·m^{-2}), temperature (T in degrees C), humidity (X_h in kg·m^{-3}) and the CO$_2$-concentration (C_a in ppm)[3].

The grower usually has the following devices available: a heating system, a CO$_2$-supply unit and a ventilation system. Optionally, growers may have: *i*) screens (*e.g.* Bakker and Van Holstein, 1995), that can either be used for energy conservation and/or shading (the latter are rarely used in tomato production); *ii*) roof sprinklers (*e.g.* Breuer and Knies, 1995) that are used for cooling and *iii*) supplementary lighting (*e.g.* Huijs, 1995) which is not used in tomato production. During extreme situations in the summer the grower may also manually spray white wash on the glass cover for shading.

The actual configuration of the control devices (and the related measuring equipment) varies between growers. With respect to CO$_2$ input, growers typically use the CO$_2$ produced by their boilers (*i.e.* only when their boilers operate on natural gas). Alternatively, some growers use pure CO$_2$. Growers who use the CO$_2$ produced by their boilers may also have a heat storage tank, allowing them to put CO$_2$ into the greenhouse while storing the hot water and use it during the night. Without heat storage capability or use of pure CO$_2$, temperature constraints will restrict CO$_2$ enrichment during the summer (*e.g.* Nederhoff, 1995).

With respect to heating some growers can use vertically mobile "growth pipes" while others have to rely solely on the ordinary tube-rail system (*e.g.* Van der Braak, 1995). Mobile growth pipes allow for local heating, for instance, near the tomatoes to stimulate ripening, or in the top of the greenhouse to stimulate growth or to compensate for night-time emittance. The ventilation system typically consists of ventilation windows on both sides of the ridge, the ratio ventilation window surface to greenhouse area varies for Venlo-type glasshouses between 0.10 and 0.23 (Waaijenberg, 1995).

The configurations of the heating system, the ventilation system together with the thermal screens and the roof sprinklers, given the fixed greenhouse characteristics, determine the spatial temperature and humidity distribution characteristics (both vertically and horizontally).

In the greenhouse one may distinguish between a global and a local climate[4]. The global climate is only measured coarsely[5], therefore, given the spatial distribution, the measured value will have some uncertainty in it. The local climate is defined as the climate directly around the individual plants[6]. It is not measured. The local cli-

---

[3] Other units also exist, for instance: light intensity in µmol·m^{-2}·s^{-1} PAR, relative humidity in % or kPa, and the CO$_2$-concentration in Pa.

[4] Also called: macro resp. micro climate.

[5] Mainly because the measuring equipment is expensive. The grower typicallly has only one or two sets of measuring devices per compartment.

[6] *E.g.* humidity near the stomata. Leaf and fruit temperature are also considered part of the local climate.

mate differs from the global climate due to limited air movement in the greenhouse and differences between the heat capacity of crop and that of the greenhouse air. As a result, the difference between what the crop perceives and what is being measured, may be significant.

It can be concluded that, owing to differences in greenhouse location and construction and differences in control devices and position of the measuring equipment, the same measurements for the set of climate variables at growers *A* and *B* do not necessarily represent the same micro climate at growers *A* and *B*. Differences in micro climate result in differences in crop response. Consequently, even with identical objectives, the operational management activities of a grower *A* and a grower *B* should be different.

2.2.2.2 The root environment

Nowadays, tomatoes are commonly grown in artificial media like rockwool slabs. Rockwool has good physical properties for plant growth (*e.g.* it is inert and has a good air-content to water-content ratio). Growers who grow their crop in rockwool slabs usually have a higher production compared to growers who produce in natural soil[7] in part because they have fewer problems with (persistent) soil-born diseases. Artificial media allow for a more precise and responsive control, however, the limited amount of water in the root environment bears an additional risk in case of malfunction.

Some growers have a root heating system, since such a system is normally not used in tomato production we will not consider it further.

The irrigation system consists of a water source[8] (usually a large basin or a well), storage tanks for nutrient solutions in concentrated form and a mixing unit. Nutrition and irrigation control involves: *i*) the concentration of the macro, micro and spore elements in the base nutrient solution[9]; *ii*) the electro-conductivity or salt content (EC in mS) and acidity (pH) of the nutrient solution[10]; *iii*) the timing, fre-

---

[7] Of course, growers producing on artificial media will also have higher production costs compared to their colleagues producing in the ground. Spaan and Van der Ploeg (1992) present an interesting discussion about the choice between producing in artificial media versus producing in natural soil.

[8] The initial water quality of the water source (especially the sodium chloride content) is an important factor in the grower's control practices. Water with a relatively high sodium chloride content can cause salt build up in the rockwool slabs, periodically the grower will try to lower the sodium chloride content of the rockwool slabs through rinsing.

[9] The grower makes concentrated nutrient solutions in two (to prevent precipitation) storage tanks. After creating the concentrated solution the proportions of the individual elements are fixed. They can only be supplied to the plants in a more or less diluted form (respectively lower or higher electro-conductivity).

[10] The grower has an "acid-tank" to reduce the pH of the nutrient solution.

quency and duration of the trickle-turns (given the fixed capacity of the drips); *iv*) the amount of oversupply. The latter three are controlled through the irrigation and nutrition control system. Currently, the grower cannot measure (nor control) the individual elements in the nutrient solution on-line. Typically, once a week the EC and pH of the nutrient solution in the rockwool slabs is measured. Every other week these samples are also chemically analyzed to determine the exact nutrient content.

The control devices for climate and nutrient and irrigation control may be controlled from one console.

2.2.2.3 Difficulties regarding controllability and uncertainty

Although greenhouse production has many advantages, with respect to controlling the crop environment, compared to agricultural production in the open air, the controllability of the environment is by no means complete.

The inability to realize the grower's settings is due to (extreme) outside conditions and the specific properties of greenhouse production. These properties are:

- the large, frequent and rapid variations in light intensity during the day (in relation to the response characteristics of the control equipment),
- the non-specificity of the ventilation and heating system with respect to individual climate variables (*e.g.* air exchange[11]), and
- the close physical relationships between radiation and temperature, and between temperature and humidity in the greenhouse.

These properties complicate the greenhouse climate control problem significantly (*e.g.* Bot, 1983; Bontsema, 1995).

With respect to uncertainty, the most apparent type of uncertainty (others are discussed later) that significantly influences the way crop production can be controlled, is our limited ability to predict the weather. Depending on the stability of worldwide current patterns, our weather can only be predicted more or less accurately 1 to 5 days in advance. Predictions of the most important variable (*i.e.* light intensity) are usually stated qualitatively and their realization will vary significantly from place to place.

The importance of various weather characteristics (*i.e.* temperature, light intensity, humidity and wind speed) for operational management varies during the growing season because of: *i*) varying possibilities for control and compensation of undesired

---

[11] Although modern greenhouses have limited leakage, the air exchange rate of a closed greenhouse varies between 0.1 and 2 times per hour (Heijnen *et al.* 1982) depending on the windspeed. The air exchange rate is important with reference to humidity control, especially during the first three months of the year.

states, *ii*) variations in the sensitivity of the crop[12] for changes in the climate variable values.

2.2.3 The production time-line

Figure 2-1 shows the characteristic activities and events during the annual tomato production cycle. To a large extent the activities and events constitute the grower's tactical plan. The timing of these milestones is an indication of a more or less standard situation; their actual timing will of course vary from grower to grower.

First, a plantbreeder sows the cultivars selected by the grower. During the first weeks the plantbreeder is responsible for proper development of the young plants. After planting the young plants at his nursery, the grower stimulates generative growth (*i.e.* truss initiation and truss development) until the third truss blooms. Following this period of generative growth stimulation, normal operational management practices commence and continue until the end of the season.

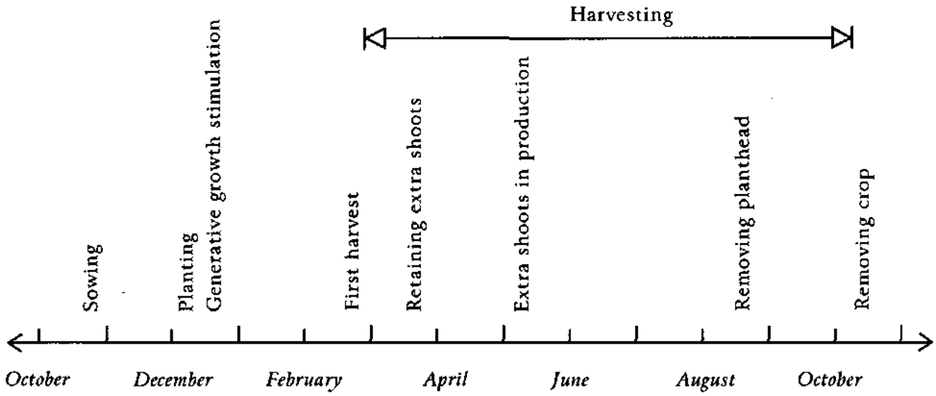


Figure 2-1 Activities and events along the production time-line.

Once or twice during the production season the grower keeps an extra side shoot[13]. This extra stem increases the number of fruits as well as the leaf area per square meter and it allows the grower to profit from the increasing light levels during spring and summer.

The greenhouse tomato is usually grown in an indeterminate way. That is, after an initial startup period, the crop keeps producing tomatoes until it is removed from

---

[12] *E.g.* a period of relatively low light levels in the winter has more impact on the growers operational management practices than proportionally the same light reduction in the summer.

[13] Typically, one side shoot per four plants (normally, side shoots are removed).

the greenhouse. Approximately eight weeks before removal of the crop, the grower removes the plant head, thereby stopping truss initiation.

2.2.4 Management and operations

Important facets in horticultural management are the grower's objectives and his activities to realize his objectives.

At the operational level, the growers' objectives are related to the following seven aspects: amount and quality of the product, timing of production, maintaining production potential, reduction of costs and production risks, and compliance to socio-economic constraints[14] (Van Straten and Challa, 1995). The way the grower operationalizes these rather broad aspects will be discussed in section 2.4.3.

The activities in a horticultural firm can be split up into two classes, the first class consists of activities that influence the crop or its environment directly. The second class are those activities that do *not* influence the crop or its environment (*e.g.* post-harvest activities like grading and packaging, or transport to market). In this thesis only the first class of activities will be considered. It is recognized that for the grower the other activities are also of importance (*e.g.* Ziggers, 1993; Leutscher, 1995).

The actions that influence the crop can be categorized into: *i*) actions that influence the root environment: activities with respect to irrigation and nutrition; *ii*) actions that influence shoot environment: activities with respect to changes in the greenhouse climate; *iii*) actions that manipulate the crop state directly: manual operations like fruit picking, leaf pruning, etc.; *iv*) actions to repress pests and diseases, *e.g.* application of fungicides or dispersal of natural enemies.

These four ways to influence the crop or its environment have generally been investigated in isolation, that is, as being separate decision making subject-matters (*e.g.* Fynn *et al.*, 1989; Martin-Clouaire *et al.*, 1993a; Van der Maas, 1991), however, interviews with growers and extension specialists show that growers tend to look at the crop and its environment in an integrated manner. Growers do not see these classes of the activities as four distinct decision problems, but consider them as a whole. From a decision making point of view, the crop and its environment must therefore be considered in its entirety.

---

[14] Regulations with respect to: use of pesticides/fungicides, labor hours of personnel, discharge of oversupply of irrigation water, etc..

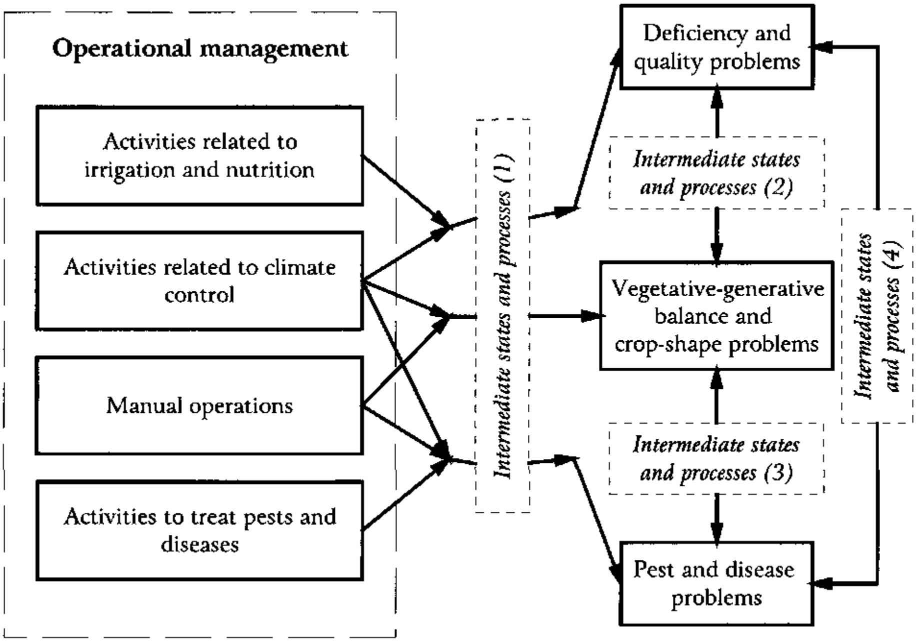


Figure 2-2 The relationship between activities within operational management (left) and key problem classes (right) in tomato production.

Figure 2-2 shows crop production from a phenomenological point of view; it indicates the activities and events that easily surface when looking at tomato production. The left-hand side shows the activities of the grower. The right-hand side shows the problem classes into which problems that occur on a tomato crop can be classified. Since quality and volume of production can be influenced substantially by problems, a grower's operational objectives are in a large part oriented towards preventing these problems. The intermediate states and processes[15] can hardly be observed in practice; we know about how activities and problems are inter-related through decades of experience and research.

The arrows between activities and problem classes indicate the most prominent interrelations between them. The grower's climate control activities are considered to be especially important because they play a prominent role in the possible emergence of problems in any of the problem classes. However, not all relations are

---

[15] The intermediate states and processes boxes indicate that plant/crop states which represent a problem are not influenced directly by activities of the grower or other crop states but through a series of processes that occur inside the plant/crop. The numbers 1 to 4 indicate that these sets of intermediate states and processes are different.

drawn, for instance, irrigation practices may influence the incidence of fungal diseases.

Figure 2-2 also shows that the occurrence of a problem in one class may cause problems in other classes, and that solutions for one problem may induce problems in other classes[16]. Since the grower is aware of the many interrelations he will approach crop production management in an integrated manner. Also, given a problem-related objective there may be several possible actions to influence the state of the crop in the desired direction. An experienced grower will have such a multifaceted viewpoint on his operational crop production management task.

The grower influences growth and development of his crop through: *i*) removing plant parts, *ii*) influencing the crop's root and shoot environment, and *iii*) influencing pest and disease populations. Thereby he uses the possibilities of his greenhouse, its control devices (as mentioned in section 2.2.1) and other (*e.g.* pest management) equipment.

The following activities can be distinguished:

1. Activities influencing the crop's shoot environment: climate control.
   In currently available climate control systems, the grower typically enters trajectories of actuator- or device-related setpoints (*e.g.* heating and ventilation temperature setpoint), state-dependent compensators (*e.g.* light-dependent increment of the ventilation temperature setpoint) and fixed device-related threshold values (*e.g.* minimum ventilation window aperture). Normally, only temperature, humidity and CO_2-concentration are being controlled. One should realize that the grower does not enter "state variable" setpoints for the three controllable climate variables but a rather complex mix of settings that result in a certain greenhouse climate.

2. Activities influencing the crop's root environment: nutrition and irrigation control.
   The grower enters setpoints for EC, pH in his control system. The timing, frequency and duration of the trickle-turns can be entered as setpoints or can be controlled based on measurements[17] or a combination of both.

3. Activities to manipulate pests and diseases directly.
   Pest and disease control involves the application of chemical pesticides and fungicides; it is also concerned with biological control. Nowadays more and more insect species and even some fungi can be controlled using their natu-

---

[16] Application of chemical means to treat pests or diseases is well known for causing growth retardation. Also, pruning (because of the wounds) promotes the incidence of fungal diseases when carried out under unfavorable environmental conditions.

[17] The grower either controls irrigation on the basis of radiation or oversupply measurements. Level tanks (Dutch: niveaubakken) measure the amount of oversupply and water uptake of three or four sample plants.

ral enemies (predators, bacteria or viruses). Natural enemies are normally dispersed manually and, when appropriate, locally near a hot spot. Chemical means (and bacteria or virus suspensions) are usually applied by spraying or fogging. Some systemic pesticides are applied through the irrigation system.

4. Manual operations.
   Manual operations are the activities: harvesting, truss-pruning, fruit thinning, branch-pruning, leaf-pruning and lowering the crop in case of a high-wire growing system (applies primarily to tomato). These activities are normally carried out in a blueprint like manner, the latter five are usually carried out once a week. Fruits are picked usually two or three times a week. Attending to the bumblebee population[18] is also considered within this class.

This section outlined tomato production, here, the grower's activities and the key problem classes have been discerned. To analyze the grower's management and problem solving activities further, some elaboration on management and problem solving literature is needed. First, some management and problem solving frameworks will be discussed (section 2.3). Then this theory will be applied to the grower's practices (section 2.4) and, finally, a model of the grower's problem solving behavior will be deduced (section 2.5).

2.3 Management and problem solving theory

2.3.1 Management theory

Management is usually considered as a cyclic process with three basic functions: planning, implementation and control (*e.g.* Boehlje and Eidman, 1984). The planning function contains the decision making activities that lead to the construction of an acceptable decision, plan or strategy. The implementation function transforms a decision, plan or strategy into executable actions. Finally, the control function consists in monitoring and evaluating and revising the implementation of a decision, plan or strategy.

Furthermore, management may be considered in relation to its planning horizon as, for instance, depicted in Davis and Olson (1985). They discriminate between strategic, tactical and operational management[19]. Strategic management is concerned with decisions that have their impact over a number of years, *e.g.* investments, the location and size of the nursery, etc.. Tactical management in greenhouse crop pro-

---

[18] Bumblebees promote proper flower pollination.

[19] The terms operational crop production management and operational management will be used interchangeably.

duction is concerned with decisions that have their impact - depending on the crop - during one or a small number of growing seasons, *e.g.* choice of cultivar, planting date, etc.. Operational crop production management concerns the day-to-day decision making processes of the grower on his nursery.

Finally, management may be considered from the grower's perspective. Spaan and Van der Ploeg (1992) use the concept of management style. The grower's management style indicates his perspective on how his greenhouse as a production facility of horticultural products can best be organized and further developed. The concept of management style has been introduced to recognize differences among individual growers. These differences influence practices within every management function at the strategic, tactical and operational level. Spaan and Van der Ploeg (1992) conclude that there is no single best way to manage a horticultural firm because each management style has its merits and drawbacks and cannot be considered seperately from the grower concerned.

2.3.2 Problem solving and decision making

Whenever decision support for a particular problem in a given organizational context is considered, it is essential to have an understanding of the problem solving process (Keen and Scott Morton, 1978). This statement may sound obvious, unfortunately it is not. Keen and Scott Morton (1978) conclude that in the study of management there seems to be lack of interest in, and ignorance of the way decisions are taken. They present a diagnostic approach to the study of decision making. The diagnostic approach requires decision support system (DSS) designers to analyze, first of all, what is happening and observe rather than presume a priori that they understand how decisions should be made.

Nowadays, information systems textbooks emphasize user-cooperation during software construction, which may indicate a change[20] in perception. However, "application conditions blindness" remains a danger that every method advocate should keep in mind. Simon (1977) already noticed that operations research enthusiasts easily forget the application conditions of their methods and elegantly solve a reduced problem that bears little resemblance with the original problem, thus making their solutions inapplicable. Without marginalizing the importance and benefits of normative approaches in, for instance, sensitivity studies (*e.g.* Leutscher, 1995), in our study a descriptive approach is considered more appropriate. Within a descriptive approach decisions are examined in terms of their process and structure. The grower's problem solving behavior will be analyzed accordingly and prescriptive or

---

[20] In the agricultural field we notice the positive appreciation received by the management styles studies of Van der Ploeg and co-workers have received and which may indicate a similar change.

normative statements regarding the decision making process of the grower will be postponed until chapter five[21].

Decision making may be analyzed according to the following viewpoints:

1. The activities of the problem owner during the decision making process.
2. The amount of structure of the problem.
3. The beliefs of certain schools of thought regarding decision making.

2.3.2.1 Activities during the decision making process

The intelligence - design - choice (and review) model of Simon (1977) is probably the most referred model for describing decision making processes. The activities carried out within each of these phases are themselves decision making processes, therefore, the process as a whole is complex.

The intelligence phase is a continuous or intermittent scanning of the environment in search for problems or opportunities. This phase results in dissatisfaction with the current state or identification of potential rewards from a new state. During the design phase of Simon's model, the problem owner, *i.e.* the grower, invents, develops and analyzes possible courses of actions. In this phase the grower has to reformulate the problem, thereby focusing on controllable elements in order to be able to tackle the problem. The final choice phase of Simon's model is the selection of the best alternative generated during the design phase, using whatever choice criteria the grower thinks are appropriate.

2.3.2.2 The amount of structure of the management problem

Simon (1977) distinguished between programmed and non-programmed decision making processes[22]. He states that decisions are programmed when they are repetitive and routine and when a clear procedure has been worked out to handle them. Programmed decisions can potentially be carried out by computer programs. Decisions are non-programmed when there is no apparent explicit procedure to handle the problem. Non-programmed tasks need the judgment of a decision maker and cannot be programmed because the underlying "deep structure", that is, the combination of objectives, trade-offs, relevant information and methodology of analysis cannot be resolved, for instance, owing to its complex or unknown nature and structure (Keen and Scott Morton, 1978).

---

[21] A decision support system, amongst others, implies a choice concerning *how* problem solving could be performed in a better way.

[22] Simon (1965, in Keen and Scott Morton, 1978) predicts that by 1975 most management tasks will be programmed. This has clearly not occurred and suggests that judgement remains essential in many decision making processes.

Bots (1989) suggests that the dichotomy between programmed and non-programmed decision making processes can be looked at from a procedure-oriented view as well as from a model-oriented view. The essence of the procedure-oriented view is the novelty of the problem in the sense that the decision maker does not know what problem-solving procedures he should follow. Alternatively, in the model-oriented view, Young (1984, in Bots, 1989) considers the degree of structure to depend on how much is known of the following components: *i*) the objectives, *ii*) the controllable and uncontrollable variables that affect the outcome, and *iii*) the relationships between these outcome-affecting variables and the outcome itself. In the model-oriented view the procedure(s) that describes how the decision maker tackles the problem, may be (more or less) known.

Keen and Scott Morton (1978) try to clarify the distinction between programmed and non-programmed decisions. They use the intelligence - design - choice cycle and suggest three types of decisions, namely unstructured, semi-structured and structured decisions. They define a problem to be structured when every phase in the decision making process is structured (note the recursiveness in this definition). An unstructured problem is a problem: *i*) where no definite procedures and conditions exist to recognize all possible problem instances within the overall management problem (intelligence), *ii*) where no methodology to solve each problem instance is available once the problem instance has been identified (design), and *iii*) where no clear criteria for choosing a best solution among those that have been created are available (choice). A semi-structured problem is a problem in which one of the toplevel phases has sufficient structure to allow for effective computer support.

Keen and Scott Morton (1978) also argue that the most valuable payoff for analytic and computer-based decision aids lies in meshing the machine's efficiency in programmed functions with the individual's judgment in unstructured problems. The questions now become: "What is meant by sufficient structure?" and "Who is going to determine that?" It is clear that these questions are themselves, at least in part, unstructured and that there is no clear-cut methodology available to tackle them.

The amount of structure of a decision problem is not a stable attribute because careful analysis may lead to more knowledge, and possibly the identification of sub-problems that have sufficient structure to allow for automation. In this light, the analysis presented here may increase the amount of structure that we ascribe to the operational management task of the grower.

2.3.2.3 Views on decision making

Keen and Scott Morton (1978) discriminate five schools of thought concerning decision making namely: *i*) the rational manager view, *ii*) the "satisficing" and process-oriented view, *iii*) the organizational procedures view, *iv*) the political view and *v*) the individual differences perspective. They stress that there is no self-evidently right way to look at the decision process and a combination of points of view may be appropriate for a given situation.

The *rational view of decision making* is highly normative and assumes a priori complete knowledge of alternative courses of action and a system of preferences or utilities to evaluate them. Many managers state that this view is much too simplistic and there is virtually no descriptive support for its conception of decision making (Keen and Scott Morton, 1978). However, this view is still extensively used, especially in studies in which only economic criteria had been used and where cases and scenario's are being presented under the banner of "the optimal solution". In the agricultural domain this approach remains popular, see for instance the studies of Huirne (1990), Leutscher (1995) or Van Henten (1994). Although the proposals within this viewpoint have difficulty in demonstrating their practical benefit[23], they can give insight in a problem, for instance, through showing the sensitivity of the output parameters with reference to changes in the controllable parameters[24].

The *"satisficing" and process-oriented view* attempts to move closer to a more realistic understanding of decision making. It states that decision makers do not strive for optimal solutions but are happy with good enough solutions. Decision makers use heuristics as a compromise between the demands of a decision and their capabilities and commitment to it (Keen and Scott Morton, 1978). Decision processes (*i.e.* the heuristics that are being used in it) can be improved but the cost for doing so, that is, the increased effort of the problem owner, will in general be too large. A decision support system (DSS) may provide a cheap and acceptable solution to improve the decision maker's heuristics (especially when these heuristics are rather simple).

The *organizational procedures' view* and the *political view* on decision making (Keen and Scott Morton, 1978) stress the procedures needed and the bargaining during decision making (*i.e.* especially in large organizations). They do not contribute much in understanding the operational greenhouse production management problem. Horticultural enterprises are typically very small and the grower is usually the only decision maker, hence strict procedures, negotiation and bargaining are not needed. These views will therefore not be discussed further.

The *individual differences perspective* considers the decision makers and their personal strategies and abilities. Spaan and Van der Ploeg (1992) show that growers differ in their general goals, that is, their attitude towards being a greenhouse grower and how they feel they should manage their enterprise. Growers, like all humans, will differ in their specific abilities with respect to the phases in decision making and their ability to process information. Keen and Scott Morton (1978) argue that DSS designers should not force all types of decision makers into one style of information processing but should give each type the kind of interaction possibilities he is psychologically attuned to. To some extent, the currently available (climate) control

---

[23] Some proposals (*e.g.* Leutscher, 1995) do not seek practical implementation while other clearly do *e.g.* Huirne (1990) and Van Henten (1994).

[24] For the part that has been modelled. It is generally recognized that non-linearities complicate the use of models in which problems have only been modelled partly.

systems have this valuable attribute[25] and this attribute should be maintained in future computerized support systems.

2.4 Operational crop production management in practice

The attributes of operational crop production management may now be discussed in the light of the principles and ideas discussed in the previous section. As already stated, the analysis will be based upon the grower's present way of working and the environment in which operational management is carried out. The boundary between tactical and operational management is somewhat vague, therefore this boundary will be defined as follows. In tomato production the following decisions are considered part of the tactical management task: *i*) decisions related to cultivar selection; *ii*) determination of sowing and planting date; *iii*) when to start with the harvest, when to keep an extra shoot and when to finish production; and; *iv*) tactical considerations related to the use of pesticides.

To a certain extent, the above decisions reflect the management style of the grower (Spaan and Van der Ploeg, 1992). Also, owing to the inherent uncertainty in the greenhouse production system, some of the above decisions may be reconsidered or adjusted according to the actual state of the crop, because these decisions are in part state-based[26].

In the following sections, the intelligence, design and choice activities carried out within crop production will be discussed.

2.4.1 The growers crop observations

An important phase in the decision making process of the grower is observing the crop's growth and development.

Observing the state of the greenhouse - crop system does not exclusively imply direct visual observation. Although visual observation is very important, growers also observe through sensing and smelling, moreover, they observe and analyze the records of crop states they have been keeping. Growers observe the crop's physiological and morphological state and progress. The crop's morphological state is described by slowly changing crop characteristics (*e.g.* flowers, fruits, plant head) that cannot

---

[25] It should be noted that the large set of possible settings in present control systems did not arise by itself but are primarily the result of demands of growers. Whether the present collection of setings is well balanced, well presented, coherent and self-explanatory regarding the relations between the settings is another matter.

[26] *E.g.* when a crop looks unhealthy near the end of the season the grower may decide to finish production earlier.

be corrected once they are developed. For the grower it is important to assess the morphological crop state as soon as possible to be able to adjust its development if this is necessary. The physiological state of the crop is described by state variables that exhibit diurnal variations (*e.g.* produced assimilates, water potential). Physiological state changes are, by definition, not permanent.

Growers tend to be very keen on what seem to be the crucial morphological characteristics of the crop. They look at the young stem just below the plant top, the orientation and the rotundity of the developing truss (important during the winter), the color of the flowers, and the color and shape of the newly formed leaves.

Furthermore, growers measure and record a number of crop characteristics, like height, length of new leaves, number of fruits set, the blooming flower, the truss that is being harvested, etc. on a weekly basis. These measurements are compared with previously collected data to determine and evaluate the rates of change. An important characteristic derived from these measurements is the fruit load (*i.e.* the number of fruits[27] present per square meter).

The records guide the grower's operational management, for instance, when a grower observes too strong a decline in the number of tomatoes per square meter over a few weeks he may be able to take precautions to remain above the critical crop load level. From interviews it became clear that growers can observe some morphological changes like the thickness of the stem just below the plant top with a coarseness of a few days (± 3 days). It means that with respect to morphological changes growers are always a few days behind.

Moreover, growers also observe certain aspects of the physiological state of the crop. Epinastic-like phenomena like the color and "curlyness" of the plant top are the most prominent indicators. The morning and late afternoon are important moments for observing the physiological state of crop. In the morning, the crop should look "fresh". For growers a fresh looking crop indicates that the crop is well watered and all assimilates have been "processed". Inappropriate physiological crop states can be a sign of, or a cause for problems[28] with the crop's morphological development. Except from water uptake, automated measurements of crop states are still in their infancy. Although a number of measurements are technically feasible, they are not used in practice. This might be due to cost and maintenance con-

---

[27] Under Dutch circumstances during the late spring - early summer (May/June), the fruit load must be at least a certain number of fruits per m^2 otherwise the crop shape is likely to deteriorate owing to a low sink/source ratio. This type of deterioration is manifested as a form of foliar deformation and is called "Short Leave Syndrome (SLS)" (Nederhoff *et al.*, 1992). The minimum number of fruits per m^2 depends amongst others upon the variety, approximately 50 for beef tomatoes but obviously much more for cherry tomatoes.

[28] A dark looking crop in the morning (the sign) may indicate that assimilates have not been processed fully. This may negatively influence the vegetative - generative balance.

straints; current practices show that calibrating sensors is a much neglected activity. Conceptually, measuring crop characteristics like the water-balance (*e.g.* Van Ieperen and Madery, 1994) can be helpful for management practices because they offer insight in fast crop processes and potentially allow for early problem detection.

Finally, growers also observe the crop's shoot and root environment. The environment of the crop is assessed to prevent possible future physiological and morphological problems. As already stated growers do not only look at the values on the console of the climate and irrigation and nutrition control systems because these values do not represent the complete picture. They also sense and smell[29] the climate situation inside the greenhouse. They try to deduce present and future inside conditions from: *i*) these observations, *ii*) the settings of the control devices and *iii*) the measured and experienced outside climate conditions. During the appraisal of a problem, growers sometimes estimate the wetness of the rockwool slabs by feeling it.

2.4.2 Domain-related terminology

Growers have developed their own jargon that they use to convey the state of the crop and the crop's environment during their discussions. Probably because they are themselves the implementors, growers tend to state their interpretation of crop processes in terms of control actions that can almost directly be carried out with the currently available control systems. For instance a grower stated: "if I see pointed trusses I should (have) increase(d) the vapor pressure deficit". Their language is often based on human analogies, they state for instance: "the crop shows a happy blush", "a dead climate", "keeping the crop active", "stimulate the crop", "the crop has done nothing during the night" or "the crop has worked hard"[30] (Schotman, 1989; Spaan and Van der Ploeg, 1992).

Unfortunately, during the discussions with growers and after reviewing professional literature, it became clear that these terms are not unambiguous in their meaning; they mean (slightly) different things to different growers. These terms are not clear because growers have difficulty in: *i*) explaining exactly what they mean (in terms of well established scientific theories), *ii*) pointing out the causes of the observed phenomena, and, *iii*) determining beforehand what precise actions to perform when a phenomenon appears.

A frequently used expression for example is: "the crop must be activated". There seems to be no unique translation in terms of processes to stimulate or actions to

---

[29] The air in a greenhouse compartment with limited air movement smells differently.

[30] In Dutch: "het gewas heeft een blije gloed", "een dood klimaat", "het gewas activeren", "het gewas heeft niks staan doen vannacht" en "het gewas heeft hard gewerkt".

perform. Growers generally state that it implies stimulating crop growth (≈ processing of assimilates), however their explanation of how the crop's "activeness" can be improved, points more to actions that raise the crop's transpiration rate.

Therefore, in the design of a computerized support system (presented in this work) only terminology for which a sufficiently reliable model can be formulated, will be used[31]. It is assumed that growers have sufficient notion of "scientifically" defined terms like transpiration, ripening, growth etc., although their models of these processes may not entirely correspond to the scientific ones. Rather ambiguous terms like "activate" or "stimulate" may also have a diplomatic connotation: when one grower advises a colleague to activate his crop he does not advise him *precisely* what to do, so when something goes wrong, he cannot be blamed (the same argument can be made for recommendations stated in professional literature).

2.4.3 Operational objectives of a typical grower

The seven important aspects that have been identified earlier (section 2.2.4) constitute, when stated normatively, in very general terms the (obviously conflicting) set of high-level objectives of growers: *i*) high amount of product with *ii*) excellent quality, while *iii*) retaining maximum production capacity, put to market at *iv*) the right time, against *v*) little costs, *vi*) without any risk and *vii*) complying to all socio-economic constraints.

The way a grower goes about these aspects, and in what manner he makes them operational and resolves the conflicts for his particular situation will be discussed here.

As an entrepreneur, a grower's predominant consideration must be the continuity of his horticultural enterprise, however, this point of departure cannot be observed in the grower's day-to-day objectives. Based on an interview with an extension specialist, growers can, at the tactical level, be grouped into two classes: the first class wants to be early on the market and tries to benefit maximally from generally higher prices early in the season. The second class is more cautious and emphasizes the generation of a proper overall crop state[32]. The growers in the second class share the opinion that a proper crop state allows for more flexibility in their control prac-

---

[31] The other choice would be to find the most proper meaning (and related actions) of growers terms. The most important reason for not pursuing this approach is that growers terms often implicitly contain objectives, which may not be appropriate outside the context of that particular grower.

[32] This distinction will - among others - result in a higher average temperature for the early growers. In brief, higher temperatures stimulate fruit growth and development (De Koning, 1994).

tices and makes controlling growth and development later in the season much easier[33].

From a normative perspective, one would expect growers to have operational "*optimizing* objectives". These objectives should be tangible and much more specific than the ones mentioned above. They should be expressed in terms of crop characteristics or crop processes that growers explicitly strive for or try to stimulate during their daily management practices. The interviews and a review of professional literature resulted in only two of such objectives, namely: *i*) (tomato) growers strive for a good vegetative - generative balance; *ii*) growers generally try to raise photosynthetic rate by increasing the CO_2-concentration[34]. Some growers explicitly strive for a good fruit taste, because they sell their tomatoes under a special brand. This objective usually results in specific irrigation practices and could be considered a third optimizing objective[35]. At present, this objective only applies to a small group of growers, however, there is a tendency that this is changing, but for most growers sufficient fruit taste is generally realized automatically, by conforming to the minimum "blueprint" settings.

Most objectives were found to be stated in a "*preventive*" or risk control manner, for instance, growers would like to prevent all kind of disorders and diseases. Instances of these classes of objectives will be worked out in more detail in sub-sections of this section.

2.4.3.1 Objectives related to crop shape problems and disorders

Crop shape or vegetative - generative balance problems are generally caused by inappropriate average temperature and/or an inappropriate day - night temperature difference (DIF) over a longer period of time. Owing to the buffering property of a crop for the "crop shape" attribute, inappropriate temperature regimes must be maintained for a number of days to see their effect. Crops may be too heavy or too thin or suffer from specific problems like Short Leave Syndrome (SLS). Preventing

---

[33] Production figures tend to support this behavior, the extension specialist reported that generally only 2 of the top 10 growers in May can keep their ranking throughout the season.

[34] An extension specialist reported that early growers sometimes (late April, early May) intentionally do not raise the CO_2-concentration. The reason behind this behavior is that they want to maintain a not too high assimilate supply (to demand ratio) thereby possibly preventing SLS in the early summer period. This is in accordance with Nederhoff *et al.* (1992).

[35] The market (*e.g.* UK market) may ask for specific fruit attributes like size (Cockshull and Ho, 1995) or keeping quality. Because of increased competition we notice a trend that these demands become more important in the Netherlands and will increase the number of "optimizing" ojectives.

an inappropriate crop shape or striving for a proper vegetative - generative balance[36] is one of the growers main operational objectives[37]. Their basic assumption is that a satisfactory crop growth balance provides the basis for a high production. The growers tend to speak about the crop "being in the right rhythm" when the balance between vegetative and generative growth is how they like it to be. Growers consider the disorder of Blossom-end rot (BER) and the crop shape problem of SLS very important. Disorders may be related to the buffering capacity of the crop. For the grower they indicate a state in which the crop has lost its "rhythm"[38] or can cause a crop to loose its rhythm. Growers consider the problem of getting the crop back into "a right rhythm" difficult. When a crop has lost its rhythm disorders like BER may reoccur more easily on later trusses.

Crop shape problems and disorders can be classified with respect to the conditions under which they are most likely to occur. The following conditions with some of the more prevalent problems associated to these conditions are discriminated:

- low light conditions (January, February and beginning of March) vegetative - generative balance: too thin; losing a truss; weak trusses
- moderate light conditions (March, April and beginning of May) vegetative - generative balance: heavy crop; fruit size too large
- high light conditions (end of May, June and July) vegetative - generative balance: SLS; BER
- high temperature conditions (July and August) vegetative - generative balance: SLS; uneven ripening; BER
- high humidity, moderate light conditions (end of August, September, October) fruit skin crazing; gold specs.

This list shows that growers vary their focus on specific problems related to these conditions throughout the year. The above problems can be considered complex[39]

---

[36] De Koning (1994) argued that growers should keep a "lean" crop, that is, a crop that does not spend more assimilates on the vegetative part than the amount that is strictly necessary for remaining productive. This amount is believed to vary during the growing season.

[37] Morphological problems can have a relative long lasting effect, *e.g.* when a grower loses a truss this may influence his management practices for six to eight weeks, since this is the average growing period of a truss.

[38] Generally, disorders can occur under extreme (outside) climate conditions (*e.g.* a high light intensity) or when irrigation and nutrition management and climate management are not in accordance with each other.

[39] The listed problems are complex because they are related to balances between supply and demand and generally involve multiple processes. The easier problems (*e.g.* chilling) are largely caused by inappropriate values of a *single* environmental variable.

because they can be caused by a range of combinations of environmental variables. Except under extreme weather conditions[40], during malfunction of the control system or due to an - incidental - human error, the easier problems hardly ever occur in modern well-managed nurseries.

2.4.3.2 Objectives related to pest and disease problems

Growers generally try to prevent fungal and bacterial problems by maintaining a "strong" crop, by keeping the crop "active" and by preventing both a strongly fluctuating growth rate and a strongly fluctuating climate. Reducing the chance of emergence of diseases means: *i*) preventing condensation on the crop, *ii*) preventing a dead climate[41] and *iii*) carrying out branch or side shoot and leaf pruning under relatively dry environmental conditions.

Growers typically adopt these general guidelines but realize that they do not always prevent emergence of diseases. Early detection is therefore important to undertake appropriate (chemical and/or biological) curative actions as soon as possible (while the damage is still small).

Growers believe that preventing serious pest problems depends upon regularly observing the crop (usually while performing crop handling activities like harvesting or pruning), that is, early detection of an emerging plague is most important (*e.g.* Van der Maas, 1991). Whenever possible growers (locally) apply biological control with natural enemies like Encarsia formosa[42].

2.4.4 Decision making

Concluding from the above, it can be said that growers operational objectives are mainly related to crop states, they have a qualitative nature and the slowly evolving crop states on which they are based can only be observed with limited accuracy. The actual operational objectives of the grower are generally reflecting a reactive problem-evading behavior. Because the sensitivity of the crop to certain problems

---

[40] During an interview an extension specialist reported that under extreme climate conditions (*e.g.* very high temperature in the summer) many growers run into the same problems, which shows the difficulty of finding proper management actions under these circumstances.

[41] A dead climate is a climate state without much air movement and a relatively high humidity. This is in accordance with the Botrytis model in Kerssies (1994).

[42] In case of biological control the grower must accept a residual damage level, because a natural enemy population cannot exist without a pest population to feed upon. The relation between crop, pest, natural enemy and environment ($\approx$ micro climate) is important for the success of biological control (*e.g.* Van Roermund, 1995). In some combinations it is problematic to generate good (environmental) conditions for a natural enemy to flourish.

varies over the season, the importance of objectives related to these problems will also vary.

Growers manage their operational objectives in the following ways. Above all, growers try to prevent problems and, to do so, adhere to certain robust and well experienced management practices. These practices, often referred to as "blueprint" practices, constitute the craftsmanship of tomato growers and are generally accepted. For instance, with respect to manual operations, growers should remove side shoots every week because otherwise the side shoots become too large[43]. Regarding nutrition and climate control tomato growers respect certain setpoint boundaries like: pH > 5, heating temperature ≥ 16°C, relative humidity < 90%, etc.. Conforming to these setpoints or practices will reduce the chance of getting problems that used to be much more common.

At any given moment during cultivation, the actual settings of the control equipment for managing shoot and root environment are the result of a continuous and progressing deliberation having taken into account all specifics[44] important to the grower. The grower assumes that his current combination of settings suffices until observations indicate the contrary. Whenever a grower observes a climate, physiological and morphological state that is not yet problematic but may possibility lead to, or indicates, the possible emergence of physiological or morphological problems in the future he considers taking action(s). The grower's action(s) are based on his mental model of the observed phenomenon. This experience-based (and situation/location specific) model relates the phenomenon to implementable actions.

Because of limitations in greenhouse climate controllability (discussed in section 2.2.2) the growers frequently encounter situations that they consider to be non-ideal for a balanced growth and development of the crop. The buffering capacity of the crop for momentary fluctuations allows growers to accept this situation that they will try to compensate for at a later stage. However, morphological (seasonal) problems may occur when this situation persists over a longer period and compensation has not been feasible. This is for instance the case during periods with low light intensity in winter, respectivily high light intensity and high temperature in the summer[45].

To what extent a grower actually has to deal with the negative consequences of control limitations and suffers from these problems depends on factors like the

---

[43] First of all, keeping these side shoots longer on the plants costs production, moreover it increases the risk of getting diseases because of the large wounds when removing the (larger) branches.

[44] Some of these specifics are: tactical plan, cultivar characteristics, greenhouse characteristics, past experiences, etc.

[45] For instance as of April forward, during periods with high light intensity (and high outside temperature), the average greenhouse temperature may rise above the desired level.

buffering capacity of his crop, the attributes of his greenhouse and control devices, and whether the grower anticipates these problems and adapts his settings in advance[46]. For example, the occurrence of SLS may be prevented by keeping sufficient fruit load. However, the grower may not be able to keep a sufficient fruit load when, due to the high average temperatures, fruit ripening proceeds at a much higher rate than the grower had anticipated[47]. Another example is anticipating or reacting upon sharp weather changes, because crops adapt to their environment, problems may be prevented by applying this strategy[48].

When a morphological problem has occurred growers will try to prevent more damage. Preventing more damage may lead to rather counter-intuitive actions, for instance, in case of the occurrence of Blossom-end rot (BER). The tomatoes that exhibit BER could be removed as soon as the grower observes it because these fruits cannot be sold anymore, however, it has been noticed in practice that removing these tomatoes can promote BER on younger tomatoes.

Pests and diseases are extremely difficult to predict, therefore growers have implemented blueprint-like guidelines and follow a reactive approach that heavily relies upon early detection.

2.4.5 Heterogeneity inside a greenhouse compartment

Growers naturally have to deal with issues of heterogeneity inside a greenhouse compartment. For instance, manual operations like side shoot pruning and harvesting may lead to variations in crop growth and development. Furthermore, differences in the stand of a plant imply differences in environmental conditions, especially near the sides and the main aisle of a compartment. Consequently, these differences result in variations in crop growth and development. Where possible, growers try to address unwanted environmental conditions that relate to these dif-

---

[46] One can conclude that growing crops under conditions with limited controllability is difficult and requires expert knowledge. Among growers the saying goes that everyone can grow tomatoes in the spring (but it is much more difficult during the summer).

[47] Fruit load planning gains interest among growers. Owing to recurrent control problems during the early summer period that, as a result, restrict the possibilities for control of the (decrease in) fruit number, an increasing number of growers tend to actively plan and manage the initiation of new fruits by retaining side shoots and truss pruning.

[48] For instance, when after a period of dull weather bright weather is expected, the grower can prepare his crop for this change by settings that increase the crop's transpiration rate. It is believed that increased transpiration will stimulate root growth and thus enlarge the water uptake capacity that is needed during bright weather.

ferences[49]. Of course, the source of variations in crop growth and development is not always known or cannot be dealt with (*e.g.* shadow of the roof construction, genetic differences between cultivars growing in the same compartment).

Depending on the state of a plant (or a group of plants) the grower observes, he may consider the observation as representative (or as an indication of a threat) for the complete compartment, or he may consider it to be a local phenomenon that requires local measures.

2.4.6 Interaction with control systems: entering decisions

The decision model of the grower forms the basis for arriving at actions. Some actions imply interaction with the control systems and result in a change in the settings.

Growers differ with respect to the frequency they change the settings on their control systems. Some growers change settings only a few times a week, while others change settings a few times a day. This observation may indicate that, the first type of growers decides upon the slower changing crop attributes thus keeping a relatively stable set of settings, while the second type of growers presumably decides upon the observable physiological and environmental states. The latter group may also rely more heavily upon the lower level control options of the control systems (fixed device values, like ventilation window aperture). These lower level controls typically require frequent adjustment.

The grower enters his decisions as value assignments to variables present in the interface of the control system. This interface provides essentially two types of settings, namely, boundary and action-oriented settings. Boundary-oriented setpoints are settings that imply a boundary value to be realized, *i.e.* they set a variable to a minimum or maximum (albeit time-variant) value (*e.g.* minimum pipe temperature must be 40°C). The action-oriented setpoints imply an action whenever the (threshold) value of a control variable is crossed. Two sub-types can be discriminated: *i*) the device-related setpoints (*e.g.* heating temperature) and, *ii*) the state dependent - device-related - setpoint adjustments (*e.g.* light-dependent ventilation setpoint increase).

The algorithms that combine the settings are hardwired in the control system and only partially known to growers[50]. These algorithms preclude a possible conflict

---

[49] They may take global measures *e.g.* installing a screen above the aisle in the summer, or they may take local measures *e.g.* simply replacing a device that is not operating properly.

[50] The overall algorithms that combine setpoints and measurements and determine control actions are only implicitly known to growers because they are seen as trade secrets of the control equipment manufacturers. An - albeit simplified - example of how settings are combined can usually be found in the manuals. Not understanding

Continued →

between the grower's settings because *i*) boundary-oriented setpoints can always be realized[51], and *ii*) action-oriented settings can always be combined and carried out using the procedures in the algorithms[52].

2.5 Synthesis

A model of decision making activities within operational crop production management may now be formulated. This model is believed to be sufficiently general to cover the differences in decision making processes among individual growers. It is also believed to be specific enough to act as a starting point for the design of a computerized support system in later chapters.

2.5.1 The grower's problem solving and decision making behavior

When analyzing the decision making behavior of growers one notes the individual differences among them. Firstly, growers appreciate the importance of various attributes of their crop differently and, will therefore value the seven before mentioned aspects differently. Secondly, the context in which growers decide plays an important role because the characteristics of greenhouse and control equipment are different for each grower. Finally, growers have different cognitive abilities with respect to decision making, moreover, some growers will be better in making observations while others may have better heuristic models for interpreting and dealing with particular problems.

Current practice shows that growers differ with respect to the frequency they change settings. Growers who change their settings frequently may - from a psychological point of view[53] - prefer to work with the lower level controls (*e.g.* fixed

---

the workings of parts of an algorithm may restrain a grower from using certain variables (especially the state-dependent compensators).

[51] The boundaries are related to the actuators themselves and have specific minimum and maximum values *i.e.* there exists a maximum temperature for the heating pipes, the window aperture must be less then 100%, etc..

[52] The algorithm computes a threshold value that logically determines a possible change in the state of an actuator. For instance: `if` air temperature > 21°C `then` the window aperture must be 30%. In this example the threshold value 21°C for the variable air temperature is computed on the bases of a certain set of action-oriented settings. Measurement of the air temperature in the greenhouse compartment determine if the window aperture must indeed be changed.

[53] Possibly because they do not completely understand or trust the decision making activities of the control systems. Also, they may have difficulty keeping track of the complete set (>100 per compartment) of settings and predicting the greenhouse climate and crop responses in advance. However, this type of grower needs to be on

Continued →

device settings). On the other hand, growers that keep a more stable combination of settings are presumably using the higher level controls like device-related setpoints and state-dependent compensators. In this case the need for change is less imperative because the setpoints for the control devices change automatically with weather changes. The present control systems, with their great number of settings, support differences in the growers' ability to process information because they allow for different levels of interaction. The large number of settings is often seen as a negative aspect (*e.g.* Bot, 1995). However, this collection of settings *does* allow for different possibilities for interaction, which is a valuable attribute from the viewpoint of the individual difference perspective (discussed in section 2.3.2.3).

The grower's problem solving and decision making behavior is analyzed against the intelligence - design - choice model of Simon (1977). During the intelligence phase we see that the grower's reaction is based upon observations of the crop and its environment. The grower's objectives are primarily stated in a qualitative and preventive manner.

During the design phase the grower has to reformulate the problem in terms of actions that can be carried out. Davis and Olson (1985) state that generating alternatives is an important part of the decision making process. Stressing the importance of generating alternatives may be caused by the normative or rational viewpoint they implicitly adopt since generating alternatives is only useful when there is an accurate and discerning procedure for the choice phase. On the basis of discussions with growers and other experts it is concluded that growers do not work out possible courses of actions in detail, but only sketch them in broad outline and consider only a few courses of action. Growers concentrate on a very limited number of key characteristics that are relevant for the problem situation (sections 2.4.1 and 2.4.3). What kind of courses of action the grower considers also depends on what attributes of his crop the grower regards as important (Spaan and Van der Ploeg, 1992).

The final choice phase of Simon's model is, in the context of the grower's decision making activities, rather trivial in the sense that only a few courses of action are being considered. The criteria upon which one course of action is being chosen are circumstantial, qualitative and are part of the expert knowledge of the grower.

The grower's way of working is in a large part caused by the uncertainty in the problem domain, that is, limitations with respect to: *i*) the accuracy of observations, *ii*) the knowledge of processes that occur in the crop and environment, and *iii*) the (in)ability to predict precisely future courses of action.

---

the edge constantly because settings must be frequently adapted to weather changes. It is sometimes questioned which way of control results in more profit. It is said that the more "restless" type of growers take better advantage of the possibilities particular weather situations offer on the other hand they will probably also make more mistakes. However, given his cognitive abilities, a grower may have difficulty changing his practices, therefore, it is doubtful whether the above question is a relevant one.

Concludingly, the way growers (should) carry out the different phases can only be stated in general (procedural) terms. It is not feasible to be rigorously complete, as there are no definite and complete sets of procedures available that can be carried out during the intelligence and design phases.

In the light of the procedure-oriented view on the amount of structure of the operational management task of the grower (section 2.3.2.2) this chapter shows some insight in the nature of this task. However, in the light of the model-oriented view, insight in the amount of structure of the operational management task remains limited, and may only be expanded through further research in the relationships between the variables and processes in the domain.

The diverse production conditions among growers and the variety in their personal preferences regarding crop production will probably not allow for the disclosement of common *operational* objectives that may - without further judgment by the grower - be used in computerized support.

The above analysis results in the conclusion that the top-level intelligence - design - choice phases of the operational crop production management problem are at most semi-structured and it is therefore not possible to fully automate one of these phases. Nonetheless, this conclusion does not imply that additional computerized support is impossible, since there may be areas within the top-level intelligence - design - choice phases that themselves have sufficient structure to allow for computerized support.

2.5.2 A model of operational crop production management

Figure 2-3 outlines a decision model of operational crop production management[54]. It includes both the activities of the grower as well as the activities of his control systems.

The left-hand side of Figure 2-3 shows the attributes involved in the intelligence phase of the decision making process of a grower. The grower appraises the current state of the crop and/or the environment[55] in the light of his objectives and expectations about the future (mainly the weather and its possible consequences for crop development), he will also have the current settings of his control system (*i.e.* the climate and irrigation and nutrition control systems combined) in mind. The intelligence phase concludes with the satisfaction or dissatisfaction with the current state. In case of dissatisfaction, the grower will have a specific objective in mind to work on. Of course, this objective is not seen in isolation, but is the present focal

---

[54] Of course, during the growing season, decision making continuously takes place, therefore the figure must be seen as an instance at a particular point in time.

[55] We state "and/or" because it is recognized that growers who frequently change the settings of their control systems, are sometimes acting mainly upon their observations of the environment (both in and outside the greenhouse).

point given all other objectives. Usually, it will not be necessary to explain this objective because the grower is in most cases the sole decision maker at his nursery. He may therefore directly translate the objective into activities carried out by himself or his personnel, and/or, into changes of the settings of the control system (or conclude that no changes need to be made). In Figure 2-3 actions concerning manual operations and pest and disease control have been put together because the presence of pests and diseases can be considered part of the set of attributes that describe the state of the crop. The design and choice phase are lumped together because the number of courses of action the grower considers, are few. If the objective requires a change of the settings of his control system, he will take a closer look at the present settings and will assess what change(s) are most likely to contribute to the realization of the objective. These adjustment(s) will generally be small, that is, only the value of one or at most a few variables will be changed at any time.

The right-hand side of Figure 2-3 shows the course of action (*i.e.* development of the overall crop and environmental state) that results after implementation of settings and/or actions under the prevailing weather conditions. The success of the implementation will depend on the actual weather. With the emergence of control systems the grower has delegated the actual control task (*i.e.* operating the control devices) to his control system; in doing so a communication problem was created. During interaction with his control system the grower has to *translate* and *communicate* a subset of his objectives in the terminology of the control system (that is: assigning values to the variables present in his control system). Decisions that require interaction with the control system make the distinction between a decision and its implementation somewhat vague. In this thesis interaction with his control system is considered part of the decision making process of the grower and not as the actual implementation of the decision. It is important to observe that the activities of the control system can conceptually be seen as (programmed) decision making.

The climate and irrigation and nutrition control systems first combine the settings of the grower together with the fixed parameters of these systems into a setpoint for every actuator, and second, through some kind of feed-back and/or feed-forward procedure, they calculate the change of the actuator value (*e.g.* Heijnen *et al.* 1982; Bontsema, 1995; Van Meurs, 1995). The circle in Figure 2-3 indicates that this process is continuously occurring. These programmed or structured decisions are generally accepted by growers because they recognize the advantages of control systems (Van Mansom and Rieswijk, 1995) and trust their working (at least up to the level at which they use them).

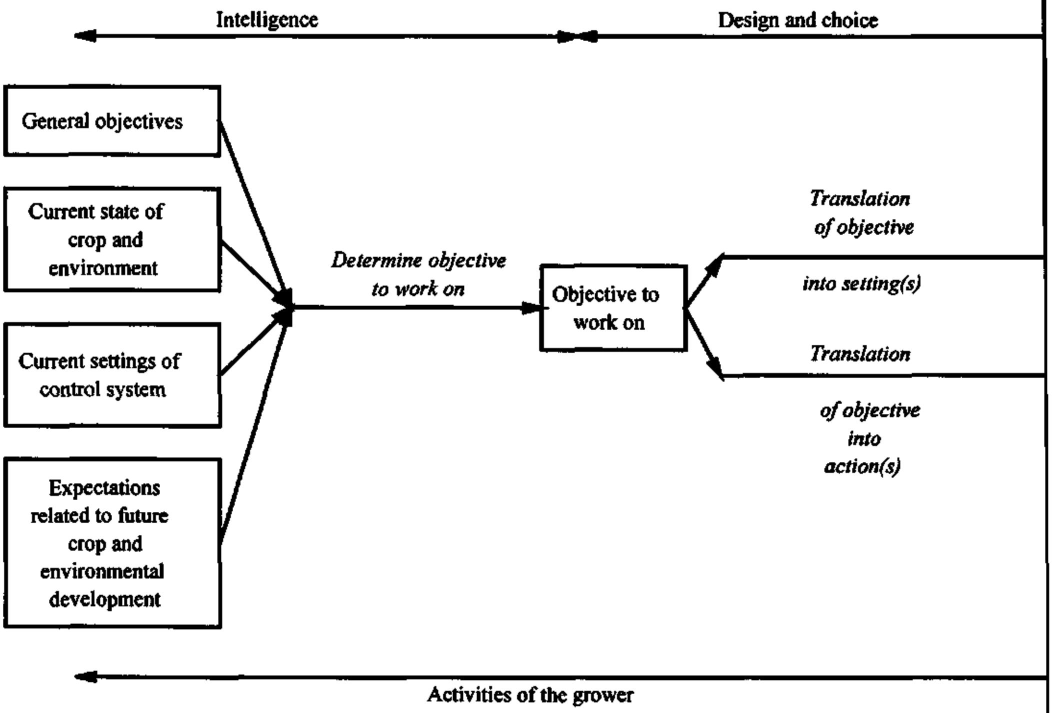
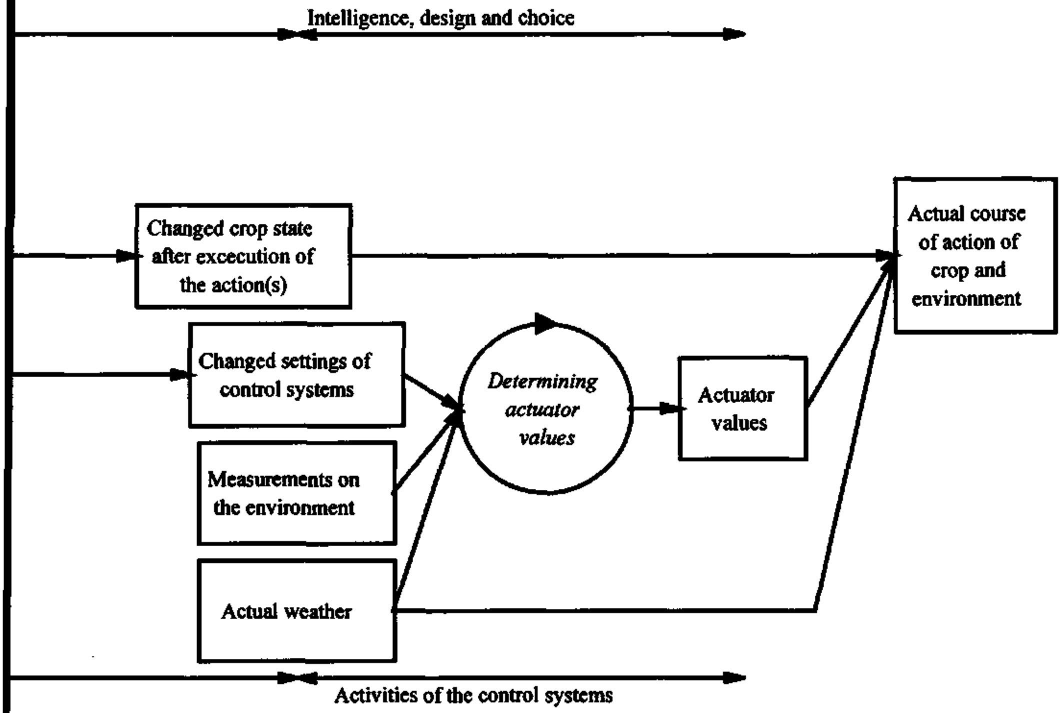


Figure 2-3 Decision making activities in operational crop production management on a horticultural firm.

The central part in Figure 2-3 is the design and choice phase of the decision making process of the grower. The grower uses practical know-how or heuristics for translating objectives into executable actions or implementable settings. These heuristics constitute his problem solving knowledge; they are necessarily qualitative, for some crop processes relatively simple and, unfortunately, for some processes even too simple with respect to the complexity of the crop process that the heuristics describe. The most important characteristic of the grower's set of heuristics is it completeness, that is, for every situation the grower will have (an) applicable heuristic(s). Some may be far from optimal (in terms of effect or payoff), but none will be destructive.



One way of improving the decisions of the grower is to enable him to use better heuristics for this difficult translation process. Replacing heuristics is precisely what has been suggested by Keen and Scott Morton (1978). (Inverted) models of processes that take place in crop and climate could be used as computerized replacements for his translation heuristics. By using such models in a computerized support system the grower can enter some of his objectives at a higher level, that is, he can enter values (or constraints) on variables that represent crop characteristics. Of course, not all aspects of crop production have been modelled sufficiently, therefore such a system should allow for interaction[56] at various levels and properly integrate the entered settings.

---

[56] Even if all aspects of crop growth would have been modelled sufficiently, some growers might still prefer to interact through variables at the actuator or climate level. Also, in order to allow for learning and a gentle shift from the present actua-

Continued →

To summarize, in such a system configuration three levels can be distinguished at which the grower can enter settings: the actuator level, the environmental level and the crop level. At each level attributes of the problem domain can be identified, *e.g.* pipe-temperature, CO_2-concentration, leaf area index respectively. Presently, the grower can enter decisions at the actuator and environmental level while his objectives are mainly located at the crop level. Entering objectives at the crop level would be a natural step forward. In part two of this thesis, a computerized support system will be suggested that operates along these lines.

---

tor and climate related variables to variables that represent crop characteristics, the first type should remain available.

3. DOMAIN KNOWLEDGE FOR COMPUTERIZED SUPPORT IN TOMATO PRODUCTION

3.1 Introduction

Knowledge in the domain of tomato production stems from many decades of experience with and experimentation on the tomato crop. This knowledge may have been written down in scientific and professional publications, but will largely reside in the minds of experts (*e.g.* growers, scientists).

In this chapter the available knowledge in the domain of tomato production will be considered from the perspective of its potential use in a knowledge-based computerized support system. For the development of extendible and maintainable knowledge-based computerized support systems it is important to have sufficient insight in the intrinsic attributes of the available pieces of knowledge (*i.e.* models) and the relationships that exist between these pieces. Without such insight one may for instance be tempted to combine certain models while application conditions of these models may not support such a combination. The objective of this chapter is therefore to present an *overview* of the characteristics of the body of knowledge available in greenhouse crop production[57]. Once such an overview is available, the usefulness of certain pieces of knowledge can be determined much easier .

This chapter has been organized as follows: first some definitions of knowledge and models are given and certain basic properties of knowledge and models will be discussed. Hereafter, the knowledge sources that make the domain knowledge available are discussed and the adequateness of domain knowledge that stems from these sources will be discussed in the light of its prospective use. Then, the representation of knowledge will be discussed. This issue is important since the choice for a representation method has a significant impact on how the knowledge can be used. Next the knowledge in the domain of tomato production will be organized using three criteria, each arrangement provides a different view on the knowledge available in crop production. Finally, the issue of missing domain knowledge will be discussed, and an example will be given of how the body of knowledge available for knowledge-based computerized support could be extended.

---

[57] In this description special emphasis will be directed towards knowledge in tomato production. We emphasize that this overview will not contain a detailed description of the available knowledge since such a description would reach beyond the scope of this work.

3.2 Knowledge and models: some definitions

Harpers dictionary (1976) describes knowledge as: "the fact or condition of apprehending truth or fact" or "the fact or condition of knowing something with familiarity gained through experience or association". *Domain* knowledge refers to "all" knowledge in a specific field or area, in this case greenhouse crop production and more specifically tomato production.

A general definition of a model is given by Minsky (1965), he describes a model as: "An object *A* is a model of object *B* for observer *C* if the observer can use *A* to answer questions that interest him about *B*." In this definition one recognizes that a model serves a purpose in the light of some use. Models appear in many shapes and sizes, for instance: maps, scale models, analogies and computer models (Kettenis, 1990). Additionally, written descriptions like chapter 2 or perceptions in the mind of the grower should also be seen as models. A model represents a part of the overall body of the domain knowledge, that is, one can construct models of pieces of knowledge.

The development of a model to be used in a computerized support system requires among others formalization and implementation[58] steps. Formalization describes the process of making the relationships between entities in the model unambiguously explicit by expressing them in an adequate representational format (*e.g.* lookup tables or differential equations). The latter may then be implemented in a particular computational framework on a computer.

With respect to the knowledge in greenhouse crop production two situations can be discerned: *i*) a model is already available in some kind of formal, possibly implemented representation[59], *ii*) a model has to be constructed on the basis of informal knowledge available from one or more knowledge sources, measurements and/or otherwise acquired data. In the latter case, the process of creating a valid computational model is usually difficult and requires, depending on ones ambition level, substantial effort.

3.3 Attributes of domain knowledge

A number of intrinsic properties can be attributed to the models exemplifying knowledge available in the domain of tomato production. These properties can be helpful in determining the usability of these models in the light of a predetermined concept for computerized support and/or a contemplated knowledge representation framework. Also, once the available domain knowledge has been analyzed, the re-

---

[58] Formalization, representation and implementation belong to the initial activities in model construction (*e.g.* Elzas, 1984).

[59] Whether such a representation is readily usable is another matter that will be discussed later.

sults of such analysis can be helpful during the selection of a suitable representation framework given the tasks and responsibilities of the computerized support system envisioned.

3.3.1 Characterizing knowledge by way of its acquisition

Knowledge-based computerized support for tomato production requires the availability of knowledge about important entities and objects apparent, and processes taking place in a tomato crop and/or its environment. Because crops are not man-made, knowledge of how crops grow and develop cannot be founded on first principles[60].

In the construction of computational models Elzas (1984) considers three approaches that may be seen as fundamental in the model building process. These approaches differ w.r.t. the way they value and exploit a priori knowledge and (objective) observational evidence. These approaches are: the phenomenological, the inducive and the deducive approach.

The phenomenological approach leads to the construction of models that are solely based on objective observations. A (stochastic) procedure is used to relate the values of the "input and output" variables in the same manner as they have been observed in the real system. Phenomenological models lack causality and assume no other a priori knowledge than the experimental data.

Models constructed using the inducive approach are entirely based on a priori knowledge, and, as such real systems are considered imperfect mappings of some higher-level principle or theory. Observations that do not agree to these hypotheses are seen as stochastic uncertainties (although they may eventually lead to additional hypotheses).

Finally, the deducive approach lays between the two former ones. It accepts a priori knowledge only insofar as it is supported by observational evidence. After an initial model has been established one uses a constructive technique to improve the model on the basis of the difference between model outcome and observations on the real system.

In crop production, the phenomenological and inducive approaches usually precede the deducive approach, and as such, the availability of phenomenological, inducive and/or deducive models to some extent indicates the present status of our domain knowledge and how far our modelling efforts have already progressed. In this con-

---

[60] Explaining the behavior of the whole based on the attributes of its parts. Although in theory this is possible for man-made artifacts, literature on model-based diagnosis (*e.g.* Hamscher *et al.*, 1992) shows that in practice it can be difficult to realize.

text deducive models are believed to better capture our present understanding of certain processes than phenomenological or inducive ones[61].

The above approaches can also be used to clarify the some of phrases often used in literature on modelling in crop production. For instance, the distinction between descriptive and explanatory (*i.e.* mechanistic) models. Explanatory models explain a phenomenon or process in terms of its mechanisms or underlying sub-processes (*e.g.* Thornley and Johnson, 1990); in descriptive or regression models, in- and outputs are directly related. Explanatory models are clearly the product of a(n ongoing[62]) deducive approach while descriptive models stem from a phenomenological approach.

In the field of artificial intelligence the terms 'deep' and 'shallow' knowledge are used. In general, these phrases refer to the same distinction as the terms explanatory and descriptive knowledge. Deep knowledge usually requires multiple levels of "reasoning" to relate inputs to conclusions (Hayes-Roth *et al.* 1983). It has been argued that deep knowledge allows for a more flexible and robust knowledge representation than shallow knowledge. Unfortunately, it has the disadvantage that structuring, maintaining and using such a knowledge base can be more difficult (Plant and Stone, 1991). Also, its use may be computationally too expensive. The question of which type of models may best be used in applications cannot be solved in general; it essentially depends upon the characteristics and goals of the application at hand. An important problem in knowledge-based systems is usually the shallowness of knowledge in the knowledge base.

3.3.2 Declarative versus procedural knowledge

Knowledge can be procedural and/or declarative with respect to both its content and its representation. The difference between a procedural and a declarative model regarding its content is important for its use and its prospect for integration with other models. Plant and Stone (1991) describe declarative knowledge from a behavioral point of view as knowledge about how actions are related to effects. Giar-

---

[61] Additionally, inducive models *may* better capture our present understanding of certain processes than phenomenological models. However, since validation is by definition impossible, this depends on the justification of the relationships contained in the model. The interested reader may want to consult Latour (1988) on this subject.

[62] Consider *e.g.* the process of photosynthesis, its results can be observed but what is really occurring we cannot (yet) tell because there is no "ground truth" that can be used to evaluate our knowledge. A part of the scientific community unravels the processes defined earlier at increasing levels of precision and detail. This decomposition approach has a deducive nature and has let us to study sub-processes within photosynthesis (like the transfer of electrons) which take place at the molecular level with a time constant of approximately 10^{-12} seconds.

ratano and Riley (1989) take a more fact-oriented viewpoint and describe declarative knowledge as knowing something (*e.g.* a statement or a relation) to be true or false. In practice the absoluteness of the latter definition may be relaxed, through probabilistic or possibilistic qualifications (*e.g.* Dubois and Prade, 1986; Klir and Folger, 1988; Pearl, 1988). Procedural knowledge is defined by Giarratano and Riley (1989) as knowledge that describes how to do something[63], that is: a procedure or sequence of actions to realize a certain goal. Plant and Stone (1991) state that, in principle, declarative knowledge is more flexible in its use than procedural, because it allows for a choice based both on the applicability and the expectations of the outcome of possible actions. This statement only applies to cases where both types of knowledge are applicable, *e.g.* in expert systems that 'do' something. Procedural knowledge does not include factual knowledge and, consequently, one may conclude that declarative knowledge can be used in a much broader context.

3.3.3 Qualitative and quantitative knowledge

Our knowledge about processes or relationships between entities in the world varies regarding aspects like accuracy, or level of detail at which knowledge is available.

The lowest level of detail considered, is the recognition that a relationship between entities exists, without knowing the functional dependency between the entities. The next level of detail could be called the qualitative level. At this level, Forbus (1984) considers the following classes of relationships between entities: *i)* inequalities (*e.g.* $X>Y$), *ii)* proportionalities (*e.g.* X increases when Y increases), *iii)* correspondences (*e.g.* X is zero when Y equals "a_qualitative_value"), and *iv)* influences (*e.g.* X decreases when Y is positive). The precision of qualitative knowledge further increases when orders of magnitude, relative proportions or linguistic values ("very low", "moderately high", "high", etc.) can be identified for the entities and the relations between them. The final level of detail, is the quantitative or axiomatic theory level. At this level, the values for and relations between quantities are numerically specified. The above classes of relationships (Forbus, 1984) still hold, they only become more precise, *e.g.* differential equations are a more precise form of "influences".

---

[63] With respect to procedural knowledge, Plant and Stone (1991) describe the behavior pattern of a digger wasp, that, after capturing a prey, carries it to the nest. It puts the prey at the entrance, enters the hole where it has previously laid an egg. The wasp then comes out of the hole and drags the prey into the egg chamber. However, when the prey is moved while the wasp was in the hole, it will repeat the whole procedure: dragging the prey to the entrance, entering the hole without the prey, returning to drag the prey down. By repeatedly moving the prey it is possible to keep the wasp in an endless loop, the wasp has no knowledge how his actions relate to the effects, it just carries out the procedure.

Qualitative reasoning (using qualitative models) has been an active research area within the field of artificial intelligence[64]. Qualitative models can be used to provide foundations for common sense reasoning, *e.g.* in second generation expert systems (*e.g.* De Kleer and Brown, 1984). Qualitative models may also be appropriate as man–machine interface because qualitative values seem to be a natural manner of communication for humans.

In greenhouse crop production one may observe that much of the knowledge is only available qualitatively; among others it resides in the minds of growers, it has been written down as recommendations in professional literature, or it can be found in (the introduction and discussion sections of) scientific publications. Unfortunately, in greenhouse crop production, there have hardly been any attempts to model knowledge using the available qualitative techniques.

3.3.4 Uncertainty and imprecision

In computerized support systems that carry out problem solving tasks in an uncertain environment it is required that uncertainty issues are explicitly addressed. What uncertainty management techniques are appropriate should be determined by the particularities of the task at hand. Krause and Clark (1993) classify uncertainty from an AI perspective and from their classification the following definitions are adopted:

- *Partial knowledge* refers to the situation were one has insufficient information to establish the truth or falsity of a statement or proposition, that is, one is *uncertain* as to its validity.
- *Indeterminate knowledge or imprecision* refers to the content of an item of information. Imprecision covers a number of possibilities and one is *ignorant* as to which one is in fact the case.
- *Vagueness* is closely related to imprecision, however as opposed to imprecision, vagueness includes a gradedness in the possibilities.

With respect to imprecision and vagueness one may consider two cases, namely: one in which a statement is sufficiently precise given its context (or *frame of discernment*) or one in which there is insufficient information available to be more precise (Klause and Clark, 1993). In the first case one may be able to further *refine* the statement to become more precise, however, doing so may increase the uncertainty of the statement[65].

---

[64] *E.g.* special issues on qualitative physics: AI journal 1984, 1991. A survey of application-oriented qualitative reasoning can be found in Travé-Massuyès and Milne (1995).

[65] Compare the statements "the crop suffers from a disease" versus "the crop suffers from Botrytis". When a grower discusses the issue of crop health with an expert (*e.g.*

Continued →

Models in crop production (whether they are of phenomenological, inducive or deducive nature) must be considered as partial knowledge, that is, one is always[66] to some extent uncertain about the validity of the model outcome. Moreover, for some processes in crop production there exist multiple competing models for which it is difficult to determine which one generates the best results given a certain objective.

3.3.5 Causality

Causality is a characteristic we can *ascribe* to a knowledge relation or model on the basis of a priori knowledge. A relationship is considered causal when source(s) and outcome are believed to be unambiguously related, which means that if the outcome has been observed one must also conclude that its cause(s) must have been present. Regarding procedural and declarative knowledge one may notice that declarative knowledge relations promise causality while procedural ones do not.

3.4 Knowledge sources for computerized support in greenhouse crop production

For developing knowledge-based support systems knowledge sources are needed. Besides written sources *i.e.* all types of publications in professional and scientific literature, three other sources of knowledge can be distinguished: *i*) growers, *ii*) scientists in the fields of horticulture, crop physiology, greenhouse physics and control theory and *iii*) extension service specialists, who may be considered intermediaries between specialists and growers.

3.4.1 Growers

Growers are trained to assess the current state of their crop, to infer situation–specific objectives and to decide about what entries to put into their control systems (see chapter two). Their knowledge is aimed at decision making and consists of practical know-how, which includes knowledge about what, how and when to observe and what treatments and/or setting changes to carry out to transfer the state of the greenhouse - crop system into a more appropriate one.

The grower's practical know-how is a blend of his personal beliefs regarding crop production processes and his unique production situation (*i.e.* cultivar, greenhouse characteristics, etc.) as simplifications of the only partially known "general truth" in the domain. Since every new growing season imposes one or more changes to his

---

an extension specialist) he is likely to be more specific than when he talks with a novice.

[66] Except for the - hypothetical - "base" model (Zeigler (1976) in Elzas (1984)).

individual situation (*i.e.* seasonal differences in weather[67] and changes in his production conditions[68]), his problem solving knowledge must also change.

The grower's practical know-how will be partially declarative and partially procedural (*e.g.* procedures to prevent and work on problems). Although a grower frequently uses quantitative information, his knowledge will be primarily qualitative, (*i.e.* his mental models about how processes proceed). His operational objectives are partly quantitative and partly qualitative; the quantitative target values he adopts are necessarily imprecise given the uncertainty in the domain.

An important problem with knowledge from a *single* grower is that without a priori knowledge it may be difficult to determine whether the knowledge relation that has been acquired from this grower, applies to his particular case alone, or may be seen as a general attribute. Especially in case of boundary values[69] one cannot tell whether they also apply in the context of another grower (or are higher resp. lower). Moreover, in discussions, growers frequently *express* their operational knowledge in an action-oriented manner. Consider for instance the expression: "if I discover a situation with attributes X and Y, then I perform action A". This type of expressions suggests the use of procedural knowledge since it describes or leads to a process or a procedure. However, further questioning may reveal additional "competing" statements in the sense that the grower's discovery of a situation with attributes X, Y and Z, may lead to the implementation of action B. These statements suggest an underlying declarative model of which at present only two particular cases have surfaced. The knowledge engineer who encounters such a case in a knowledge acquisition session should at least ask the grower why the particular actions should be performed, and if there are more attributes that could also be of importance. He or she may even attempt to discover an underlying declarative model in which situations and consequences are properly related.

3.4.2 Scientists

The type of research that scientists perform in the fields supporting to greenhouse crop production, determines the value of their knowledge for computerized support.

---

[67] Every growing season has its own specific problems, growers are likely to remember recent experiences better than older ones. It is also conceivable that some growers perform better in mild than in hot summers (and vice versa).

[68] For instance: *i*) the control device configuration (new actuators, *e.g.* heat buffer tank, screens, changes in the heating pipe arrangement), *ii*) the root environment (*e.g.* foam rubber slabs instead of rockwool slabs), *iii*) the cultivar, *iv*) socio-economic constraints (*e.g.* additional constraints regarding the use of pesticides, price of natural gas, irrigation water recirculation requirements), *v*) quality constraints (*e.g.* regarding taste or size), *vi*) timing of planting, etc..

[69] "If X stays above 'a' then problem P will not occur".

The following scientists can be of value: *i*) researchers specialized in the fields directly supporting horticultural production: *i.e.* plant and crop physiology (plant-water-nutrient relations, growth and development physiology), entomology, phytopathology, physics and control science (*i.e.* the parts of the latter that relate to greenhouse production) and horticultural economics, and *ii*) researchers specialized in the tomato crop.

The knowledge possessed by these specialists is usually declarative, explanatory and not specialized for a specific production situation (although some of it may only be applicable to certain geographic areas *e.g.* northwestern Europe). Their knowledge has partly been laid down in scientific papers that have undergone extensive review. This review has increased the level of confidence in this knowledge. Unfortunately, only a small part the researchers' body of knowledge is already formalized using some kind of representation formalism (*e.g.* mathematical equations).

Researchers may have extensive knowledge of specific processes taking place in a tomato crop, however, they usually miss the necessary overview of the entire production process and are less familiar with the particularities of it. As a result their decision making capabilities will be limited.

Owing to the way scientific knowledge has been acquired[70], its declarative, explanatory and non-specific nature, this body of knowledge is relatively stable and less brittle than the knowledge of growers.

3.4.3 The extension service

The extension service specialists are the intermediaries between researchers and growers. They have a good notion of the key problems in tomato production because they frequently interact with growers. Their task primary has been the translation of the results of scientific research into practical advice in an applicable format for the grower. Extension service specialists often advise growers on problems in their nursery. Consequently, they must be capable of observing the situation in the greenhouse and assessing the value of situation–specific problem solving measures. Because extension service specialists come into contact with a wide range of production situations they have good insight in the particularities and subtle differences between them. Therefore, they are generally skillful in assessing the pro's and con's of a specific action in the light of the complete production process.

The knowledge of extension service specialists is in part laid down in professional literature. Some of these writings may be seen as translations of scientific publications. In these writings some of the details, generalities and uncertainties that were present in the scientific publications have been removed or placed in a certain context more appropriate for the target group readers to consume.

---

[70] As part of the more or less generally accepted working practices, scientists tend to be as careful and complete as possible in their explanation of cause and effects.

3.4.4 Written knowledge sources

Written knowledge sources range from general textbooks on how to grow tomatoes (*e.g.* Moerman *et al.* 1986) to scientific articles detailedly describing aspects of processes underlying crop production (*e.g.* photosynthesis). While the majority of written knowledge sources contain qualitative informal knowledge, it can - as opposed to the above sources - also contain formalized (quantitative or qualitative) knowledge in the form of model equations, rule bases or lookup tables. This formal knowledge may also be implemented in some kind of programming language.

3.4.5 What source to use

The determination of the best source to use depends on the activities for which the source is used. From a knowledge acquisition perspective all sources are useful, albeit in different phases in the knowledge acquisition process. Growers are the best source to inform about which states, variables/entities and processes are important (*e.g.* chapter two). Extension service specialists can provide the knowledge engineer with initial (conceptual) models of how processes and variables relate to each other. Furthermore, they are able to validate the statements made by the growers. For instance, they can distinguish between knowledge that implicitly contains situation–specific characteristics and knowledge that expresses generally valid relationships in crop production. Finally, scientists can validate and extend the models that are generated with the help of the extension service specialists. They are also a good source to determine the level of detail that is appropriate given the requirements set forward by the knowledge engineer. Moreover, scientists can advise about the use of adjustable parameters for identification purposes.

Written knowledge sources can be used in all stages of the knowledge acquisition process, that is, sometimes as background information and sometimes as a source for a formally represented model.

3.5 Knowledge representation

3.5.1 Formal representation schemes

A formal knowledge representation scheme may be defined as a representation mechanism that can be used to unambiguously specify a piece of knowledge. Although one may (sometimes) unambiguously specify small pieces of knowledge in natural language, natural language is not considered a formal representation scheme because of the possibility of differences in interpretation. Informally represented knowledge exists in the minds of humans and can also be found in written publications.

The process of formally specifying knowledge is intentional, that is, one plans to be formal. Generating a formally represented model out of an informal representation requires effort. Depending on ones level of ambition this effort may be substantial, possibly requiring multiple knowledge sources, additional knowledge acquisition, and experimentation with the newly generated model (*e.g.* Elzas, 1984; Zeigler, 1984; Schreiber *et al.*,1995).

Nowadays there exist numerous formal schemes to represent knowledge. Many of the presently available representation schemes stem from the field of artificial intelligence (or its predecessors). They can for instance be found in artificial intelligence (AI) textbooks (*e.g.* Rich and Knight, 1991; Winston, 1992).

Some of the more common representation schemes are: *i*) mathematical equations, *ii*) logical assertions and rule bases; *iii*) semantic nets, frame-based representations and object hierarchies; *iv*) constraint networks; and *v*) neural nets. In this list, a system of mathematical equations is by far the most commonly used scheme to represent a model.

Rich and Knight (1991) state that with respect to the representation of knowledge there is much controversy about which framework is better; unfortunately, there is no clear-cut answer to this question in general, but (nowadays) specific rational choices can be made for specific applications[71].

3.5.2 Procedural versus declarative representation

Procedural and declarative aspects can be observed when looking at the representation[72] of knowledge. Rich and Knight (1991) state that a declarative knowledge representation is one in which knowledge is specified but the use to which that knowledge is put is not given. To use this knowledge we must augment it with a program that specifies what is to be done with the knowledge and how. Rich and Knight (1991) define a procedural knowledge representation as one in which the control information that is necessary to use the knowledge is considered to be embedded in the knowledge itself. To use this knowledge only an interpreter that "understands" the instructions given in the knowledge[73] is needed.

---

[71] Among others, the following parameters are of importance: available budget for knowledge acquisition, the required accuracy of the output of the model, the representation framework in which the model must be integrated, etc..

[72] Here we change our viewpoint and leap from the knowledge to the symbol level (Newell, 1982), at this time we do not consider connectionist approaches like neural networks.

[73] For instance, a PROLOG program (which is a procedural representation of the logical assertions), or a simulation program like SUCROS87 (Spitters *et al.*, 1989).

Declarative knowledge allows - in contrast to procedural knowledge - for much more freedom in its representation. Declarative knowledge could be represented in both a declarative as well as procedural fashion, since representation and content are essentially independent. Moreover, declarative knowledge does not suggest a particular direction of reasoning[74]. The direction of reasoning will depend on the possibilities of inference engine[75] and/or the way the model relations are formulated. As an example consider, for instance, the simulation program TRANSMOD (Stanghellini, 1987) which describes the transpiration process of greenhouse crops. TRANSMOD is a procedural representation of a number of declarative equations between entities in the problem domain[76]. When one decides to use the knowledge contained in TRANSMOD, one may also consider to use this knowledge in some other, more appropriate declarative or procedural format given the task at hand.

With respect to procedural knowledge one does not have this freedom since it can only be represented using a procedural scheme and the direction of reasoning is fixed.

3.5.3 Qualitative and quantitative representation

Since there is much variation in the precision of pieces of domain knowledge, an advantage of an approach in which quantitative and qualitative knowledge can be combined, is that a piece of knowledge can be represented at its most appropriate precision level. Using a combination of quantitative and qualitative knowledge requires some method to integrate them. A common method is the use of mapping functions. Some entities in the knowledge base may both have a real number representation and an aggregated symbolic one. The mapping function uses a partitioning of the real line to determine qualitative values from quantitative ones and, quantitative intervals from symbolic values[77] (*e.g.* Travé-Massuyès, 1995; Guerrin, 1991). Whenever both qualitative and quantitative representations exist, the quali-

---

[74] Two directions can be distinguished: the forward inference direction and the reversed inference direction (or goal or treatment-oriented direction). The forward inference direction is sometimes seen as the causal direction. However, since processes and corresponding state variables are *defined* attributes of a system, the causality relations are inherited.

[75] Some representation schemes (*e.g.* constraint reasoning) allow for reasoning with a model in both directions.

[76] The model relates the transpiration rate to the CO_2-concentration, light intensity, leaf area, etc.. In comparisons (Jolliet and Bailey, 1992), TRANSMOD proved to be one of the most accurate transpiration models for greenhouse crops .

[77] The mapping process can also be carried out by using fuzzy sets.

tative value may be seen as an imprecise representation in the light of the quantitative one[78].

3.5.4 Representation of uncertainty and imprecision

An important aspect in the representation of knowledge is the issue of uncertainty. Much of the need to include uncertainty in a representation framework stems from the fact that models are often incomplete. This is especially the case in diagnosis tasks (in which one tries to find a cause for observations). Another important reason to represent a notion of uncertainty is in tasks that involve prediction in situations that possibly allow for multiple courses of events.

Some of the more common methods to represent uncertainty and imprecision are: *i*) methods based on Bayesian statistics: certainty factors, Bayesian networks and Dempster-Shafer theory; *ii*) methods based on possibility theory and fuzzy sets (*e.g.* Dubois and Prade, 1985; Pearl, 1988; Krause and Clark, 1993).

In greenhouse crop production some types of uncertainty may best be represented with statistical methods while for others fuzzy sets may be better suited[79]. In theoretical studies on control applications in greenhouse crop production (*e.g.* Van Henten, 1994; Tap *et al.*, 1996), uncertainty issues are circumvented (and partly ignored) through frequent recalculation on the basis of the most recently available prediction.

3.5.5 Reuse and integration

Reuse of already formally represented knowledge requires that this knowledge is accessible. In the field of agricultural crop production many numerical models have been generated and are in principle available for reuse (*e.g.* CAMASE, 1996). Unfortunately, reuse is not easy and a number of attributes that relate to representation issues impede reuse in applications. Two of these attributes are: the model is only available as an executable computer program or as source code[80]; and second, speci-

---

[78] One should note that not every qualitative value is necessarily imprecise, if no quantitative counterpart exists a qualitative value is as precise as it can get.

[79] *E.g.* uncertainty in expected weather can be represented by using statistical methods, while preferences of the grower may be better represented by using fuzzy sets (*e.g.* Martin-Clouaire *et al.*, 1993b).

[80] Many model representations are unfortunately specified in this way: *e.g.* TRANSMOD (Stanghellini,1987); TOMSIM (Heuvelink, 1996). There are several advantages of representations like ESP diagrams (*e.g.* Elzas, 1986) or mathematical equations as compared to source code: *i*) They are easier to read by humans, *ii*) they do not suffer from programming bugs and do not depend on the programming capabilities of

Continued →

fications, application conditions and underlying assumptions are not available or not clear enough[81]. It is clear that practitioners would greatly benefit from easier and more lucid representation, in addition, other researchers may benefit even more.

To make matters more difficult, even if a model is already available in some kind of formal representation, it must comply to additional demands to allow for integration with other models in a specific knowledge representation scheme. For instance, the model must be representable in the chosen format, it must not be in conflict with knowledge already entered, etc.. Possibilities to combine models into an integrated framework largely depend on the representational and inferential adequacy of the framework itself (Rich and Knight, 1991). Although changes in the representation of a model are far less difficult than changes in the intrinsic attributes of some body of knowledge (which essentially amounts to acquiring more knowledge), they may still require substantial effort and insight.

3.6 Organizing crop production knowledge

To clarify the relationships between entities taking part in the different processes in the problem domain some kind of criterion to organize knowledge about tomato production may be helpful. Three possible arrangements will be discussed here. Each of these arrangements considers the knowledge available in crop production from a different perspective. These arrangements also implicitly present the important states and processes in the domain of greenhouse crop production (without explaining the details involved).

3.6.1 Reaction time

Processes and related state variables in the problem domain are commonly arranged according to the response time of the processes (*e.g.* Bakker *et al.*, 1995). The rates of change of entities that may be considered relevant in the light of computerized support vary considerably. Four classes may be discerned: *i*) instantaneous biochemical processes, *ii*) fast processes with a time step in the range from minutes to hours, *iii*) relatively slow processes with a time step in the range from hours to days, *iv*) slow developmental or long term processes with a time step in the range from days to months. The boundaries in this arrangement are chosen on practical grounds and do not imply a strict theoretical separation. In general one could say

---

the researcher (who is usually an amateur programmer), *iii*) they guard against over-enthusiastic use.

[81] They may be described in a narrative style and are for instance spread throughout a (Ph.D.) thesis (*e.g.* Stanghellini, 1987; Heuvelink, 1996).

that the lower the degree of aggregation of a process, the smaller its reponse time will be.

One may argue that this arrangement is somewhat arbitrary since the processes that are being classified, have been defined by ourselves, consequently, their time constants are also defined by us. Second, the time constants of these defined processes are sometimes difficult to determine especially for the processes that are considered to progress slowly. The different categories contain the following (classes of) processes:

1. Instantaneous bio-chemical and physical processes
   Intra-organ processes that are sub-processes (*e.g.* electron loading/unloading) in aggregate processes like *e.g.* photosynthesis. Since these processes are of limited importance in the management task of a grower they will not be discussed further.
2. Fast processes with a time step in the range from minutes to hours
   Photosynthesis; transpiration; water uptake; nutrients uptake, transport and distribution of water, nutrients and carbohydrates; stomata behavior; processes belonging to the mass and energy balances in the greenhouse climate (including transfers between local (boundary layer) and global climate).
3. Relatively slow processes with a time step in the range from hours to days
   Growth (assimilation) and dry matter distribution; growth (size or volume increase), development and ripening of the fruit (including problems like multi-colored fruits); development of certain deficiency and abundance problems (*e.g.* blossom-end rot, gold specks, cuticle cracking); root growth (water uptake capacity); leaf area development (single leaf); truss initiation; pest and disease development (at the individual level).
4. Slow developmental processes with a time step in the range from days to months
   Fruitload development; leaf area development (crop); development of pests and diseases at the population level (*e.g.* spider mites, white flies and leaf miners, respectively Botrytis, Fusarium and Verticillium); fruit (quality) development.

3.6.2 Chain of influence

A second way in which the knowledge in the problem domain can be arranged is more oriented towards variables and states (Figure 3-1). Four levels have been discerned: *i*) the crop level (at this level reside the state variables that characterize the crop), *ii*) the environmental level (the variables that describe the root and shoot environment are situated at this level), *iii*) the control and manual operations level (the variables/activities at this level are under direct control of the grower), and, finally *iv*) the external environment level (at this level the non-controllable variables are situated).

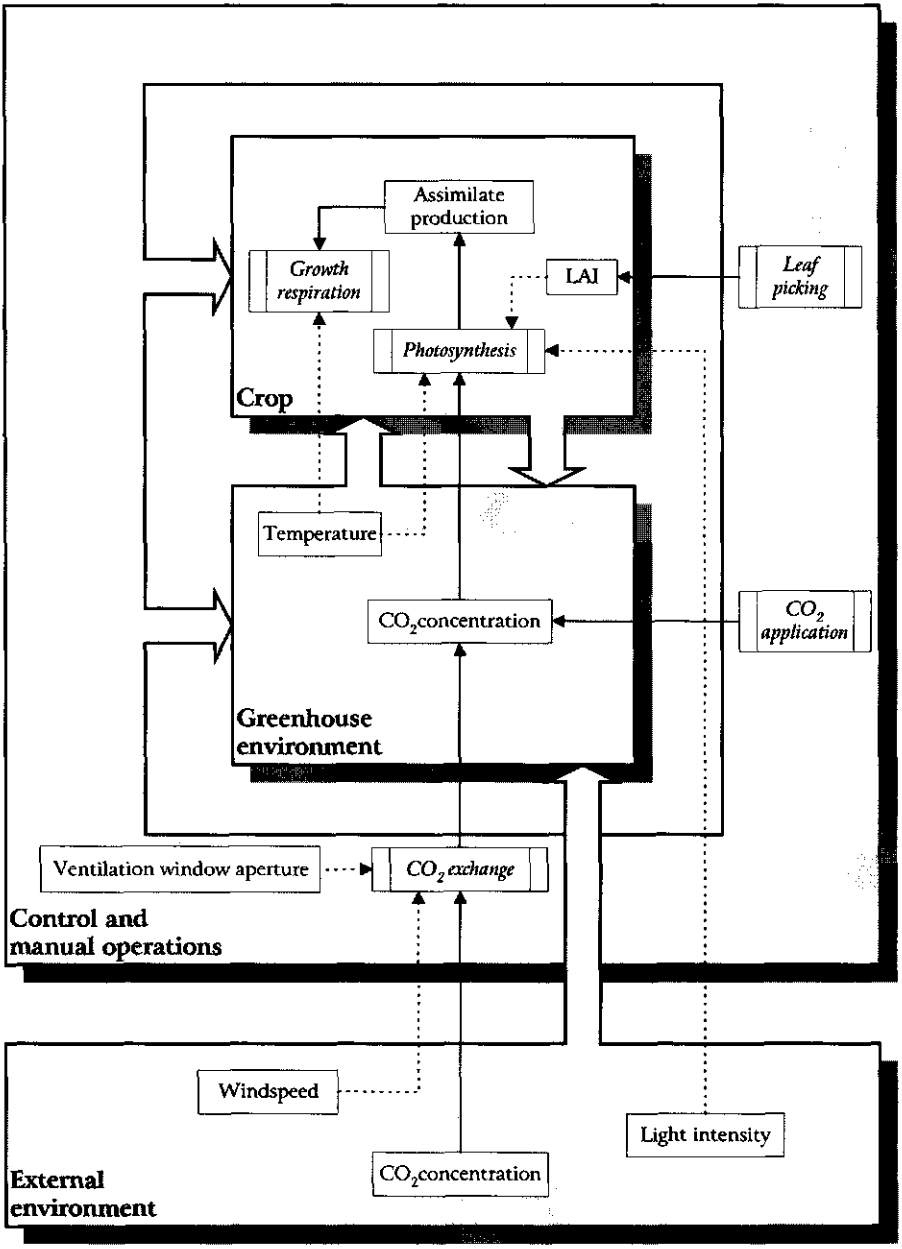


Figure 3-1 The influence chain arrangement with some states and processes.

The chain of influence in this arrangement is mainly from the external environment level to the crop level (via control and manual operations level and the (internal) environmental level). This may be observed when one considers the processes that are situated between the states. The control and manual operations level stands out because influences on crop states and processes can be direct and do not always go via the root and shoot environment of the crop (*e.g.* the influence of the pipe temperature on transpiration, or the influence of the pruning of leaves on the LAI). The different levels include the following variables:

i. The crop level
The crop level holds numerous characteristics that may be ascribed to the crop. Some of the more important variables that may be discriminated are: leaf area index (LAI); biomass; fruit ripening rate; assimilate production; transpiration rate; stomata resistance; fruit load; risk of Botrytis; fruitsize upon harvest; number of fruits per truss; 'the crop's measure of heaviness'; risk of Blossom-end rot; flower color; leaf color; tickness of the young stem; orientation of the trusses; etc.. Most of these variables require additional spatial (*e.g.* per m^2, for fruit number x) and temporal (*e.g.* per hour, per week) specification. Some variables like 'the risk of Botrytis' describe virtual characteristics that can only be 'guestimated'. Some of these characteristics may be expressed numerically while most can only be described qualitatively[82].

ii. The root and shoot environment level
Average air temperature; vertical temperature distribution; humidity; light intensity above crop; vertical light extinction; CO_2-concentration; wind speed/air movement; air exchange rate; pH rootzone; EC rootzone; water-air ratio (rootzone). Growers sometimes use the term 'dead climate', this phrase may be considered as a aggregate variable partly describing the overall shoot environment.

iii. The control and manual operations level
Depending on the control devices available to the grower, the following variables may be situated at this level[83]: pipe temperature; temperature growth pipe; place growth pipe; ventilation window aperture (lee and luff side); CO_2-application; roof sprinklers; screen aperture; pH irrigation water; EC irrigation water; irrigation frequency; irrigation oversupply; individual nutrients concentration and proportions. With respect to manual operations we may distinguish the following variables: number of leaves to be picked; average number of fruits harvested; harvest frequency; fungicide application frequency; remaining number of fruits after pruning; etc..

---

[82] Some of the phrases used by growers are concise descriptions of the overall state of the crop *e.g.* 'a happy looking crop'. It is unlikely that this type of phrases can ever be modelled objectively.

[83] One may argue that a number of the fixed greenhouse characteristics also belong in this category since they also determine the resulting root and shoot environment. These characteristics are: orientation, height, cover type and roof angle of the greenhouse, and the kind of growing medium.

iv. The external environment level
   Light intensity; sun angle and azimuth; cloudiness (ratio direct - diffuse light); temperature; humidity; wind speed.

This arrangement emphasizes that the variables that relate to the grower's objectives and the variables that influence these objectives reside at different levels. The former reside at the crop level while the latter reside at the other three levels.

The first two arrangements combined show that most of the crop attributes are influenced by slowly progressing processes. The rates of change of these processes together with the intrinsic buffering capacity per variable determine the variation per time unit and, consequently, the necessary control alertness (of the grower).

3.6.3 Hierarchical decomposition

The hierarchical decomposition arrangement has (among others) been suggested by Thornley and Johnson (1990). They use a hierarchical decomposition that is based on scientific reductionism. Their organizational levels include: crop (i+1), plant (i), organs (i-1), tissues (i-2), etc.. They state that the understanding of processes at the i^{th} level is based on component processes at the $(i\text{-}1)^{th}$ level (and possibly at lower levels). In mechanistic models that simulate the behavior of processes at the crop level, the plant level is sometimes skipped, in this case, the processes at the crop level are explained by their component processes at the organ level[84].

A direct consequence of the idea that component processes can be used to describe higher level processes is the necessity to have so-called 'pools' of intermediate compounds . Pools of compounds connect the respective component processes and allow them to progress at different reaction speeds and/or allow them to have different maximum capacities[85]. The use of component processes together with the concept of internal pools of compounds is attractive because many growth and development processes in a crop (including problems like disorders or deficiencies) can be explained through it. Unfortunately, the concept imposes methodological problems during modelling since, in general, the internal pools of compounds are (very) difficult to measure. Although Sinclair and Seligman (1996) advise against the use of 'hypothetical' internal pools of compounds, De Koning (1994) and Heuvelink (1996) showed that such a concept (*i.e.* sugar pool) can be used to model dry matter distribution in tomato. Heuvelink (1996) also showed that the concept of sugar pool can be experimentally confirmed.

---

[84] This is for instance done in some transpiration and photosynthesis models.

[85] Some of the component processes can be seen as 'supply' processes while others can be seen as 'demand' processes. In some cases the supply process will be relatively fast while the demand process is relatively slow (this for instance applies to the relationship between photosynthesis and crop growth), in other case it may be the opposite. An example regarding capacities of greenhouse vegetable crops is the difference between the crop's water uptake capacity and its higher transpiration capacity.

The concept of internal pools of compounds and its related supply and demand processes may be used to structure the knowledge in the knowledge base of a computerized support system. Using a structurization that implicitly represents an essential characteristic of the domain knowledge has the advantage that modifying and augmenting this knowledge base is easier (Waterman, 1986).

3.7 Concluding remarks

3.7.1 The availability of domain knowledge in tomato production

When overviewing the literature about processes that play a role in the growth and development of greenhouse tomatoes (see the arrangement in section 3.6.1 for a listing of the most important ones), one may conclude that the relatively fast processes are known in much more detail than the slowly progressing processes. To a large extent this may be caused by our inability to measure or simulate certain important internal states in plants[86] which could function as inputs for the models simulating slowly progressing processes[87]. Additionally, and in particular with respect to pests and diseases, differences between global and local environmental conditions[88] contribute to our inability to model the processes underlying the development of these pests and diseases.

Although we have considerable knowledge about greenhouse crop production processes, much of it and, especially knowledge about processes that are most interesting to growers (like the development of diseases and deficiencies), is only available in a qualitative and informal form. It is therefore noteworthy that attempts to formalize parts of this body of knowledge using qualitative modelling techniques (*e.g.* Forbus, 1984) have been very limited. To investigate the modelling of a slowly progressing crop problem, and to gain more insight in the applicability of qualitative

---

[86] *E.g.* the osmotic and water potentials at certain places in a plant, see Van Ieperen (1996) for a modelling attempt.

[87] An example in which science has succeeded in accurately modelling a relatively slow process is crop growth (dry matter production). Important input for the crop growth model is the state variable CO_2-uptake which is affected by the photosynthesis process.

[88] Foremost this refers to the shoot environment but it also applies in the root environment. There seems to be little interest in modelling the local (boundary layer) climate within a canopy. Some studies have been carried out (*e.g.* Goudriaan, 1977) but they have not yet resulted in a model that is suitable for practical use. However without better insight in the conditions in this boundary layer, accurately modelling fungal development (*i.e.* Botrytis) will remain problematic.

modelling techniques, the problem of Blossom-end rot will be modelled qualitatively in Appendix 1.

From an application point of view it may be advantageous to use a combination of quantitative and qualitative knowledge instead of either quantitative or qualitative knowledge alone. In contrast to the use of only quantitative knowledge[89], a mixed approach allows much more domain knowledge to be used and may lead to a more versatile knowledge base. Of course, such a mixed approach requires a representation and inference method that can deal with both types of knowledge in an integrated fashion.

3.7.2 Research versus application models

Sinclair and Seligman (1996) and Young *et al.* (1995) remark that there are important differences between the use of models as a research tool and the use of models in applications (*e.g.* in decision support or process control).

Numerical models for scientific discovery, often still based on a reductionism approach, need to - in the light of the hypotheses being tested - adequately simulate the ***behavior*** of the state variables under consideration. Absolute agreement between the calculated and observed values is of secondary importance and, as Elzas (1984) and Pease and Bull (1992) point out, unattainable given the enormous number of simultaneously operating physical and chemical processes. This notion, which implicitly shows the limits of the reductionism approach, is regularly addressed by experts in the field (*e.g.* Elzas, 1984; Young *et al.*, 1995; Sinclair and Seligman, 1996).

The use of models in applications requires that the values of the variables simulated in these models have sufficient[90] precision since these values are used as a foundation for decisions[91]. A number of examples have shown that detailed mechanistic models are not necessarily the best choice in applications since they may be outperformed by much simpler models (*e.g.* Sinclair and Seligman, 1996; Young and Lees,

---

[89] Not using qualitative techniques may result in a knowledge gap that requires other means to resolve. In chapter four it will be shown that many approaches follow a purely quantitative approach and try to fill the knowledge gap by introducing state constraints that represent unmodelled phenomena that are considered important.

[90] The level of accuracy needed is in decision support often an application dependent social construct, that is, users of the model will eventually determine whether the provided accuracy is sufficient. In control applications, (simulation) experiments may be set up to determine whether the accuracy is adequate. Additionally, on-line identification methods may be used to maintain sufficient accuracy.

[91] Here, the decision support system is not considered as a learning tool. In a learning environment the requirements are considerable less demanding, that is, one can gain insight from simulation runs although the models used may be quite crude.

1996). Simplification may lead to models that are valid within certain conditions or situations. An additional advantage of these simpler models is the faster speed of execution of their implementations.

3.7.3 Completeness versus simplicity

Models used in applications should be easily parameterized preferably by the grower himself or his regular advisor (an extension specialist). It is important that all control variables be included in models for which they are relevant. Not including some of the control variables (*e.g.* the vertically mobile growth pipe) will likely have a negative effect on the confidence of the grower in the model, particularly in models that simulate states for which the grower believes the control variable is important. Hence, simplicity and completeness must be well balanced.

Since simplicity may be introduced at the expense of robustness, applications should have provisions for conditions that are outside the validity range of a model. One possibility may be the use of multiple models which have overlapping validity ranges.

Concludingly, one can state that the use of crop models under practical conditions is not a trivial undertaking. The availability of research models only indicates that utilization of the knowledge laid down in these models *may* be possible; it certainly does not guarantee success. Often significant effort is needed to rework the important properties of a research model into an implementation that can be used in a practice. Most of the models presented in literature should be seen in this light. The efforts required for developing application models explain in part why model use in horticultural practice is still in its infancy.

3.7.4 Next steps

In the section 3.6, the states and processes that are important in greenhouse crop production have been mentioned and related to each other using three different arrangements. A knowledge-based computerized support system that assists the grower in his operational management task should contain models in which at least a subset of these states and processes are present.

A natural next step could be to bring together the available knowledge about these states and processes. Given the vast body of knowledge that is already available in written form (this knowledge, if brought together, would fill a small library), it is argued that this activity would be quite an undertaking, and that carrying out such an undertaking is inappropriate in the light of objectives set forward in chapter one[92]. Moreover, without insight in the precise tasks and objectives (which are pre-

---

[92] Besides, others have already to some extent pursued this objective. For instance Bakker *et al.* (1995) give (from their point of view) an overview of the knowledge

Continued →

sented in later chapters) of the computerized support system to be developed, gathering, augmenting and formalizing domain knowledge for such a system can be an inefficient undertaking. Especially since the choice of an appropriate representation formalism has not been yet been made. It is for this reason that further acquisition and formalization of the knowledge needed is not pursued at this stage.

---

needed/available in greenhouse climate control, and, Thornley and Johnson (1990) look at physiological crop processes from a formal, mathematical point of view. Additionally, the interested reader may for instance consult: Stangellini (1987), Bakker (1991), De Koning (1994), Marcelis (1994), Nederhoff (1994), Heuvelink (1996) and Van Ieperen (1996) for detailed studies of processes in vegetable crops; Kerssies (1994) and Van Roermund (1995) for studies on diseases and pests; or De Jong (1990), Van Henten (1994) and Tap (in preparation) for recent studies concerning greenhouse climate physics and greenhouse climate control.

4. A SURVEY OF APPROACHES FOR COMPUTERIZED SUPPORT[93]

4.1 Introduction

A grower's operational management practices can roughly be split up into decision making activities and implementation functions. Parts of the implementation functions are presently supported by climate and irrigation/nutrition control systems. These systems carry out (programmed) decisions (chapter two), and, operate the actuators based on both the settings of the grower and the environmental conditions perceived.

The recent past has seen a number of attempts to resolve the problem of operational crop production management using computer-based decision support and/or decision making systems. There is little doubt that additional computerized support will become indispensable for proper (including: profitable, ecologically sound, etc.) management of greenhouse production systems in the near future.

The purpose of this chapter is to examine the main existing approaches that have addressed the problem of computerized support[94] for greenhouse production management. In this respect one may envision the following improvements.

The currently available support systems may be improved upon without changing their present responsibilities. Such improvements (discussed in section 4.2) could, for instance, lead to a better realization of the settings entered by the grower or a reduced energy consumption (or lower operational costs).

A more ambitious form of decision support would consist in adopting a computerized approach of (parts of) the intelligence, design and choice activities currently carried out by the grower in the computerized support system. Section 4.3 discusses support for the intelligence task of the grower, more specifically, providing the grower with better (diagnostic and/or predictive) information to base his decisions on.

---

[93] Parts of this chapter have been taken from: Martin-Clouaire, R., P.J. Schotman and M. Tchamitchian, 1996. A survey of computer-based approaches for greenhouse climate management. *Proceedings of the IFAC/ISHS second international workshop on mathematical and control applications in agriculture and horticulture.* Silsoe, UK, 12-15 September 1994. Acta Horticulturae 406, p 409-423.

[94] The phrase 'computerized support' is preferred over 'decision support' to denote that the supervisory systems that support the grower in his management task do more than support decision making processes. These systems also control the crop's environment and as such they carry out decisions themselves.

A number of approaches found in literature have investigated the issue of developing methods that take over (parts of) the design and choice activities of the grower (section 4.4). Some approaches propose systems for complete automation of climate or irrigation/nutrition management, that is, systems that may operate without any interaction by the grower. Most approaches are less ambitious; many embrace the notion that it may be possible to find better setpoints and/or actuator settings than the grower and his current control systems are capable of. These proposals usually present a procedure that describes how this goal may be realized. Some of these approaches (*e.g.* Martin-Clouaire *et al.* 1993a) interface with currently available control systems, and the proposed decision can in principle be accepted or refused by the grower. Therefore, these approaches are easier to introduce in the grower's practice and in the existing greenhouse computer environment. Other proposals that adopt part of the design and choice activities of the grower also imply a fundamental change in the manner in which part of the implementation functions are carried out[95] (the way in which the settings of the actuators are computed). Particularly, in proposals that are founded on the so-called "optimal control" methodology (*e.g.* Van Henten, 1994).

4.2 Attributes of currently available control systems and possible improvements of the control algorithms

The present greenhouse (climate) controllers have evolved from a simple multivariable control system based on classical feedback control schemes. In their simplest setting only two variables play a role; in the climate control case these are temperature and humidity. In practice these variables are controlled by separate control loops. Since both control loops can lead to changes in the value for the same actuator (*i.e.* the ventilation windows, under conditions that require cooling), some way of combining the outcome is necessary. Generally, heuristic rules are used to determine a final value (Bot, 1995).

Over time the complexity of the climate control system has increased[96] because more and more improvements were incorporated. Two aspects of increase in the complexity of the climate control system can be distinguished. First, increase in complexity of the control algorithm itself and, second, increase in complexity of the user interface of the (climate) control system.

---

[95] This approach aims at additional efficiency gains by arguing that striving for certain climate setpoints can be costly and may, in the light of the objectives of the grower, not be necessary.

[96] The reason for increased complexity is partly due to the growers. Over time they asked for more and more ways to manipulate the behavior of the actuators, rapid advances in hard- and software (*e.g.* increase in computational power and computer memory) allowed it. Improvements to the control algorithm *e.g.* feed-forward response light intensity changes have been pushed by the manufacturers.

The control algorithm has become more complicated because: *i*) additional variables have been introduced (*i.e.* CO_2-concentration), *ii*) feed-forward mechanisms have been built in (to allow for a smoother temperature control) and, *iii*) the grower has been allowed to interfere in the algorithm through the introduction of (state-variant) minimum and maximum boundary values for the actuators.

The user interface has become complicated because of the following.

- The introduction of time-variance, *i.e.* allowing for setpoints for different periods within a 24 hour period (typically day, evening, night and early morning) including transitional phases between these periods.
- For humidity, the ability to use multiple units[97] *i.e.* relative humidity in % and in $kg \cdot m^{-3}$.
- The introduction of state-based setting changes, most prominently light dependent (heating temperature) setpoint increase.
- The introduction of minimum or maximum actuator settings, that can vary on the basis of the outside conditions.
- Presently available control systems are not designed for specific crops. They can be used for any crop although some settings (*e.g.* day and night lengthening / shortening) are relevant for some crops and have no meaning for others, unfortunately, these settings sometimes remain visible in the user interface.

Increase in complexity of the user interface does not directly complicate the control algorithm itself, however it requires pre-calculations to determine what the actual setpoints will be. Over time, these pre-calculations for combining the grower's settings increased in complexity, and at present they may not be very transparent to the grower.

Additionally, the number of possible settings per compartment are large (typically more than 100) which further complicates the possibility for the grower to estimate the behavior of the control system in advance. Bot (1995) correctly notices that present settings bear only indirect meaning for the process to be controlled, namely crop production[98]. It is therefore logical that the grower is more concerned with the overall behavior of the control system and is less interested in the individual settings[99].

---

[97] Commercial control systems originally used different units. Therefore growers asked their vendors to make other units available also, such that they were able to compare the humidity values with their colleagues.

[98] Bot (1995) considers 'making money' the ultimate goal of the grower. In chapter two it is argued that the goals of the grower are more diverse.

[99] See the analysis in chapter two. The individual settings obviously play a role when deciding about a change, but only in the light of the overall behavior.

The changes to the present control systems have been gradual and their fundamental underpinnings are based on approximately 30 year old technology. The question is therefore legitimate whether it would be worthwhile *for the benefit of the grower* to re-design these systems using newer technologies. Without extending the scope of present control systems, advantages may lie in more effective and/or efficient setpoint tracking. The basic assumption behind the idea of better setpoint tracking is that growers specifically strive for certain (climate) setpoints. Given this conjecture[100], the idea that setpoint tracking can be improved, is based upon the following. First, with regard to efficiency, currently available climate control systems permit simultaneous heating and ventilation, obviously, this practice is inefficient with respect to energy consumption. The use of model-based methods may lead to reduced energy consumption. Second, regarding effectivity, developments in greenhouse climate modelling (*e.g.* Bot, 1983) allow the application of model-based control algorithms that outperform (w.r.t. smoothness, reaction on disturbances, etc.) the complicated heuristic-based algorithms.

Methods to improve setpoint tracking have often been suggested, however, most of them only include the air-temperature as the variable to be controlled (*e.g.* Tantau, 1985; Udink ten Cate, 1987). Young *et al.*(1987), however, suggest the proportional-integral-plus control system design which is applied to simultaneous temperature, humidity and CO_2-concentration control in greenhouses (Lees *et al.*, 1996). This approach is based upon a low order linear model of the dominant modes of the system. During a winter trial, results show smooth and accurate setpoint tracking, reduced energy consumption and reduced mechanical wear and tear on the actuators.

Systems using newer techniques may improve setpoint tracking, however, they also require growers to change their control practices. The grower will have less possibilities to interfere with the control algorithm since interfering will reduce the performance of the approach (although state dependent setting changes remain possible). Whether the possibilities for control that are left remain acceptable is something for the growers to determine. Model-based control has been applied in practice without clear success by Indal (a former manufacturer of climate control systems). Growers felt that the Indal system behaved in an unpredictable way which they did not appreciate. It is not clear whether failure should be attributed to the control method, the interface (that might not have allowed for sufficient possibilities for interaction), or to insufficient preparation and training of the growers using the system.

During the final discussion of the 2nd international workshop on mathematical and control applications in agriculture and horticulture (Silsoe, UK, 12-15 September 1994) discussions between model-based control advocates and representatives of suppliers of climate control systems revealed that these control system suppliers do not yet see a commercial advantage in replacing their present control algorithms

---

[100] Which is, as argumented in the above, not completely correct.

with model-based ones. They argued that the advantages for growers are too limited to justify the investment. However, recent extensions to their systems regarding CO_2 and energy management (Anonymous, 1995) may stimulate replacement of present heuristics-based algorithms.

4.3 Providing information to support the intelligence task

The intelligence task of the grower (see Figure 2-3) is concerned with the search for situations that require decision making. In the case of operational production management the intelligence task includes: *i*) identification of the possible presence of a problem with respect to the crop or its environment, *ii*) prediction of the possible emergence of a problem in the near future, or *iii*) the search for opportunities[101].

The intelligence task may thus be supported through additional information on the present and future state of the crop and its environment. Two additional sources of information are being considered, namely additional measurements and the introduction of simulation models.

4.3.1 Additional information through new sensors

For inside measurements the grower typically uses the following sensors: a resistive type Pt100 sensor for temperature, dry and wet bulb psychrometer for humidity, infrared analyzer for CO_2-concentration (usually in a multiplex sampling configuration). For the root zone, pH and EC sensors are widely used, some growers also use soil (*i.e.* substrate) moisture sensors (Gieling and Schurer, 1995) or level tanks[102].

Many new sensors have been proposed to ameliorate the information that may be acquired from the crop and its environment. In hydroponics ion selective electrodes have been suggested for on-line monitoring of the ion uptake (*e.g.* Gieling *et al.*, 1988; Bailey *et al.*, 1988). Monitoring the uptake of ions like Ca^{2+} or K^+ could function as an early warning system and improve the growers insight in the possible emergence of disorders. Measurements directly on the crop have also been suggested, for instance:

- De Koning and Bakker (1991) report a weighing beam that allows on-line registration of the fresh weight of one or more plants over time. This system has been applied successfully at a small number of nurseries.

---

[101] Questions like: "How can the balance between vegetative and generative growth be improved?", "How can the risk of Blossom-end rot be reduced?", "How can the heat storage tank best be filled?".

[102] Containers that can monitor the water uptake of three or four "standard" plants.

- Van Ieperen and Madery (1994) report an intricate weighing system consisting of two lysimeters. It allows for on-line measurement of fresh weight increase, transpiration and water uptake at the same time.
- Infrared temperature probes (*e.g.* Stanghellini, 1987) and evaporimeters (Schmidt, 1996) have been reported for measuring the leaf temperature and transpiration.
- Rose (1994) reports a heat-balance sap-flow gauge that can be used to monitor the water flow through the main stem. Oscillations in the water flow due to stomata closure can be observed with this system.
- Devices to measure the growth (size) of individual tomato fruits (*e.g.* Ho *et al.*, 1987) have also been suggested.

Above measuring devices typically stem from research environments where they have been applied successfully.

4.3.2 Additional information through models

The models discussed in chapter 3 can in principle be used to simulate, as opposed to measure, certain crop states. In the agricultural domain some practical successes of model use have been reported (*e.g.* McKinion *et al.*,1989).

Models have the additional advantage that they can be used in scenario studies, in which, based on expected weather, the course of certain crop state variables may be predicted. Given proper embedding, models can thus be a useful source of information, for instance, the grower may monitor the rate of change of certain state variables and possibly prevent the occurrence of undesired states in reality.

Descriptive or explanatory models may also be embedded in diagnostic systems. However, most diagnostic systems reported in horticultural literature contain heuristics-based backward reasoning models. A user typically consults such a system to diagnose a problem situation, and/or to acquire advice regarding a solution strategy. Kozai (1985) reports a tomato crop disease diagnostic system that covers seventeen diseases and four disorders. The system interactively diagnoses a problem and recommends a treatment, this recommendation is partly based on of the equipment available at the grower's greenhouse. Unfortunately, no experimental results are given. Guay and Gauthier (1991) report a tomato disorder identification system. Their prototype system contains information about six disorders. Informal trials showed successful identification of the disorders included. Unfortunately, no subsequent information could be found on this system, therefore its practical value remains unknown.

At present, the use of simulation models in horticultural enterprises for operational management support is still in its infancy.

4.3.3 Use, robustness and value of new sources of information

The practical availability of the information sources discussed in sections 4.3.1 and 4.3.2 has been limited possibly because of the following reasons:

- Models are only valid over a sub-range of the environmental conditions that occur in the greenhouse. Models are, per definition, more or less inaccurate, consequently, the modelled process diverges from its real world counterpart. Regular (on-line) parameter identification and/or feedback on states is therefore needed. Unfortunately, research in practical (on-line) identification methods for crop production has been insufficient.
- Most sensors described in section 4.3.1 are not yet considered robust enough to be applicable in practice. Frequent calibration is needed. Gieling and Schurer (1995) consider a calibration frequency of more than twice a year not to be acceptable for use in horticultural practice.
- Local crop/plant measurements ask for a representative site because these measurements are supposed to characterize a whole compartment. Frequent assessment of the measuring site by the grower is therefore needed[103]. Consequences of failing to do so can be costly because a complete compartment could be controlled on the basis of these (non-representative) measurements.
- Some sensors (*e.g.* the infrared leaf temperature sensor) are considered too expensive to be used in practice.
- Growers are known for their, rather widespread, lack of interest in sensor maintenance. However, this attitude is changing and is likely to improve if their business becomes increasingly depended upon reliable measurements.
- Suppliers of control systems have until now focused on the technical aspects of environmental control which may be logical given their background and the rapid changes in the soft- and hardware industry. They also have tried to keep the control systems generally applicable which is understandable from a commercial viewpoint, since greenhouse production can be seen as one of many niche markets. Introducing crop specific characteristics will make these systems less generally applicable and requires the suppliers to invest extensively into new knowledge domains (*e.g.* crop physiology).
- Interpretation of modelling results may be difficult and requires additional training of growers. According to an extension specialist, one may say that the successful use of the GOSSYM/COMAX system (*e.g.* McKinion *et al.*,1989) by cotton growers must in part be attributed to the substantial support the extension service provided (Childers, personal communication).

---

[103] Growers who are using level tanks already have had to deal with this problem.

Introduction of diagnostic systems for use on a farm has proven to be difficult. Developing a commercially attractive system (*e.g.* Hilhorst, 1992) is costly, while the market may not be willing to pay the price (Hilhorst, personal communication). Backus concluded with regard to an expert system for the diagnosis of bacterial diseases in mushrooms that the value of these diagnostic systems for experienced users seems limited. They may be more valuable as a teaching tool[104] (*e.g.* Backus *et al.*, 1991; Backus, personal communication). Since diagnostic systems are normally based on heuristic backward reasoning models their predictive power is limited. Forecasting systems, for instance, systems that predict the outbreak of certain problems (*e.g.* Botrytis, see Kerssies, 1994) through application of explanatory models may, in theory, be more helpful[105].

4.4 Supporting or taking over decision making activities of the grower

The systems proposed for management support differ as to the broadness and scope of the management problem they address. With respect to the four areas or domains identified in figure 2-2, the approaches reported in literature all concentrate on one single domain to influence the state of the crop (*i.e.* mostly climate management). The reasons for their proposition are manifold. First of all, an improvement in economic gain with respect to blueprint settings is being pursued (*e.g.* Bailey, 1990; Seginer and McClendon, 1992). Secondly, a reduction in the workload of the grower (Jacobson *et al.* 1987) or assistance in solving a difficult problem for him (Martin-Clouaire *et al.* 1993a) is aimed at. Thirdly, reduced energy consumption is also frequently pursued (*e.g.* Martin-Clouaire *et al.* 1993a; Tap, in preparation).

In the following sub-sections the approaches considered are arranged according to their level of ambition. The level of ambition of an approach grows with the number of activities of the intelligence, design and choice cycle that are carried out by the computerized support system.

The following two levels are identified:

1. Systems that are meant to *completely* take over the activities of the grower. These systems carry out the intelligence, design and choice activities con-

---

[104] The EPIPRE system (*e.g.* Zadoks and Schein, 1979), after an initial success, showed strong a decline in use. This decline has been attributed to user-learning. Users learned how the model inside the system 'reasoned' and could, after some time, predict its behavior. At that stage, the system did not provide enough added value and was abandoned by its users.

[105] These models may even be directly implemented in a (climate) control system (*e.g.* Lange and Tantau, 1996).

cerning the task at hand. They operate on an implicit notion of goals. In some systems the grower acts as a sensor.

2. Systems that carry out *part* of decision making activities that belong to the design and choice phases. These activities are currently the sole territory of the grower. These systems are less ambitious since they allow growers to enter certain decisions. Given his input and an embedded objective function, these systems perform some kind of optimization to determine actuator values or setpoints.

The approaches considered typically use techniques from the fields of artificial intelligence (AI) and control theory. Within the approaches that have their roots in control theory, the optimal control approach using Pontryagin's Maximum Principle (*e.g.* Seginer *et al.*, 1986) seems dominant.

4.4.1 Autonomous control systems

4.4.1.1 Approaches using techniques from the field of artificial intelligence

The idea of using rules as a repository of precompiled procedural knowledge about what to do in a given situation was first proposed by Hoshi and Kozai (1984). The management of the greenhouse climate sub-task addressed by the systems that use a rule-based approach is under the responsibility of an inference engine that must continuously select and coordinate the rules to apply depending on the current situation and the available data.

Harazono *et al.* (1988) and Harazono (1991) describe a rule-based system for climate management. This system has the objective to optimize photosynthesis. The system continuously monitors setpoints for air temperature, relative humidity and CO_2 concentration. The rules in the system describe the processes of photosynthesis and transpiration together with the mechanisms to influence these processes. The system deduces, on the basis of relative changes in the crop's response to changes in greenhouse climate, whether the crop behaves appropriately and, if not, it turns to symbolic reasoning and tries to infer new setpoint values by firing and chaining the rules in its knowledge base. Harazono (1991) reports a short (one day) and preliminary experiment conducted in a growth chamber with a lettuce crop. The results suggest the feasibility of the approach in this particular setting. The environmental changes decided by the system were found reasonable (agronomically) and gradual (no sharp changes). Apparently no large scale experiments in a real greenhouse and with a more elaborate knowledge base were planned.

4.4.1.2 Approaches using techniques from the field of artificial intelligence that include some kind of optimization

Jacobson *et al.* (1987) have implemented a system for the generation of setpoints (temperature, CO_2-concentration and humidity) for climate management of a tomato crop (see also Jones *et al.*, 1988). The system combines an optimization algorithm that uses mathematical models to calculate optimal setpoints with a rule-based part that acts as a supervisor to ensure robustness and reliability of the calculated setpoints. The optimization routine performs a seasonal optimization and calculates hourly setpoints given different weather scenario's. The sets of hourly setpoints are checked against acceptable ranges by the expert system. The rationale behind this approach is to make sure the suggested setpoints make sense, since it is realized that the models simulate only part of reality. Unfortunately no experimental results are given.

Fynn *et al.* (1989) describe a system to decide about nutrient management for a cucumber crop. The system is a combination of an expert system and a utility model which expresses uncertainty in a probabilistic way. The objective of the system is to determine an optimal nutrient recipe. At the start of each day the expected values of possible recipes are calculated based on: *i*) the recipe of the previous day, *ii*) expected radiation levels, *iii*) past inside climate conditions, *iv*) current crop conditions with respect to maturity and nutrient deficiencies, and *v*) simulation of expected crop transpiration. In this system the role of the grower is to supply information about the state of the crop. The authors report proof of principle, unfortunately, no comparison with other methods for nutrient management have been shown.

4.4.1.3 Speaking plant approaches for climate control

Normally climate control is considered from an engineering perspective (*e.g.* Bot and Challa, 1991; Udink ten Cate, 1983) but it could also be seen from a plant-physiological perspective (Hashimoto et al. 1985). The phrase "speaking plant approach" (SPA) has been introduced in Udink ten Cate *et al.* (1978). The approach is based on the idea of plant growth control using direct measurements of a "standard" plant or *speaking* plant (Takakura, 1975). Measurements which have been suggested are: leaf temperature and water-content of the stem (Hashimoto, 1980). These on-line measurements provide insight in the short-term plant response and, using a crop-specific algorithm, climate-setpoints or control actions can be derived (Hashimoto *et al.*, 1981). In Hashimoto *et al.* (1981) the leaf-temperature versus air-temperature ratio is assumed to indicate the stomata aperture and therefore provides relevant information regarding the rate of photosynthesis.

Problems with the speaking plant approach are: *i)* it is reactive and static (only the current plant state is taken into account); *ii)* determination of the control algorithm for a given measurement and its underlying relations is not straightforward. For example, leaf temperature does depend on stomata aperture but also on the incoming radiation. *iii)* for many, especially developmental processes (e.g. fruit ripening, disorder development) on-line measurement is not possible. As a consequence, the main problem with the speaking plant approach is that it is not complete, it considers only part of reality. Hashimoto (1991) therefore mentions that the approach should be expanded and knowledge (from expert-growers), implemented by means of AI-techniques, should be introduced in the speaking plant approach[106].

4.4.2 Computerized support by setpoint determining or control action generating systems

The systems described here generate setpoints or control actions on a mixed basis, that is, both on the basis of input from the grower and on the basis on an implicit notion of goals. Usually, these goals imply the maximization or minimization of some kind of objective function (*e.g.* maximization of photosynthesis or minimization of the energy consumption). The way in which the grower can express his objectives varies considerably between the approaches. Some allow the grower to interact in terms of (constraints on) climate variables (*e.g.* Challa and Van Straten, 1993) while others allow him to communicate some of his objectives at the crop level, for instance, in the SERRISTE system (Martin-Clouaire *et al.* 1993a) the grower can choose to stimulate or repress the crop characteristic "vigor".

4.4.2.1 Approaches based on optimization

Optimal control approaches for climate control are defined by the use of a numerical model of the system (crop + greenhouse), of a performance criterion (a scalar function) and the definition of a time interval over which optimal control is determined. Such approaches are young in the horticultural field because they need accurate models of both the greenhouse production system and the profit production rate. The philosophy behind these approaches may be considered to stem from two sources:

1. The notion that it may be possible to have better growing crops (≡ higher production) through better climate control. It has been shown in simulation that, in the light of some objective function, it is possible to produce better setpoints than some blueprint settings that were considered to simulate the settings of a typical grower (*e.g.* Seginer *et al.*, 1986).
2. The notion that some processes may be optimized computationally which the grower is not capable of (photosynthesis) or which the grower may need not to be aware of (*i.e.* the fast greenhouse dynamics). This notion, together with the crop's integrating capabilities with respect to temperature

---

[106] However, one may also argue that some of the worth-while properties of the SPA (see also 4.3.1) may be used in a knowledge-based environment. Both suggestions place a technique before solving the problem, in chapter five it will be argued that it may be better to start with (the characteristics of) the problem.

> (*e.g.* De Koning, 1989), has resulted in the observation that it may not be necessary to maintain certain (rigid) temperature setpoints but instead have the grower strive for average diurnal values within certain momentaneous boundaries. Consequently, momentaneous actuator steerings can now be calculated that respond much more flexibly to the disturbances and opportunities of the outside weather conditions.

The emphasis placed on either notion varies between the approaches, some mainly consider the first (*e.g.* Gal, *et al.*, 1984; Seginer and McClendon, 1992), while others (*e.g.* Challa and Van Straten, 1993; Tap, in preparation) are more interested in the energy savings that stem from the second or try to include both (*e.g.* Van Henten, 1994).

The formulation of the control problem in terms of the performance criterion, the models used, and the optimization period considered, varies considerably among the approaches. These three aspects are highly interrelated, for instance, the optimization period chosen will have its influence on the formulation of the performance criterion. Also, the method for calculating the optimal controls and the number of variables to be controlled vary among the approaches and will affect the model(s) used in the system.

To get some understanding of the characteristics and potential of the optimization approaches, they will be discussed on the basis of the following attributes:

1. Formalization of the problem
   Important differences can be observed in the formulation of the optimization problem. Essentially, the objective is to optimize climate control over a large part of the growing season thereby calculating optimal control actions or setpoints. In essence the problem is computationally expensive because of the fast disturbances (*i.e.* weather) and the long optimization period. Frequent weather fluctuations (common in western Europe) require timely calculation of control actions. Key issue is how to solve computational problems that occur in this kind of stiff optimization problems (Challa and Van Straten, 1993). One way to decrease the computational complexity is to reduce the number of state variables (see point four). Another is to ignore the fast dynamics and instead of calculating control actions, daily setpoints can be calculated. This is for instance done in Gal *et al.* (1984), Seginer *et al.* (1986), Seginer (1991) and Van Henten and Bontsema (1991). However, Tap *et al.* (1993) shows that ignoring these dynamics can seriously influence the performance. Van Henten (1994) uses a two time-scale approach in which the slow and fast sub-processes within the crop production process are separated. In this approach, based on Pontryagin's Maximum Principle, the co-states calculated during the seasonal optimization, are interpreted as the marginal values of the crop state variables to changes in the climate variables. As a result, these co-states are used in the fast sub-problem and can lead to efficient dynamic control.

2. Performance criterion

   The performance criterion represents in essence the combination of multiple, usually conflicting, objectives expressed in the same unit, namely, the financial gain per timestep. This so called profit production rate can be taken as an economic measure of the crop response to climate manipulation. It was first derived by Challa (1980) and further developed by Challa and Schapendonk (1986) and Challa and Heuvelink (1993). Because of differences in both the formulation of the control problem and the crop production conditions the performance criterion varies considerably in the individual studies. Gal *et al.* (1984) and Seginer *et al.* (1986) minimize the overall costs to generate a crop ready to market. This final state is predetermined and the period in which this state is reached will vary. Later Seginer (1991) and Seginer and McClendon (1992) maximize income (at termination) minus costs for heating and rent over fixed 30 day period, given constant prices. Gutman *et al.* (1993) minimizes the heating cost, using a limited time horizon of a few days for which weather forecasts are considered available. Instead of taking constant prices Van Henten (1994) includes a value function for the crop (lettuce) based on auction prices.

   Important is also the consideration of the unmodelled phenomena like risk of diseases, nutrient deficiencies and other considerations with respect to product and crop quality. One approach to incorporate these phenomena indirectly is through the use of so called penalty functions as an extension to the performance criterion (*e.g.* Van Henten, 1994). The penalty functions try to prevent violation of certain (extreme) boundary values for some variable (*e.g.* lower and upper humidity levels (Van Straten and Challa, 1995)).

3. Variables to be controlled

   Optimal control approaches vary in the variables for which optimal trajectories are computed. Some approaches consider temperature only (*e.g.* Seginer, 1991 and Seginer and McClendon, 1992), others consider only the CO_2-concentration (*e.g.* Seginer *et al.*, 1986). Van Henten and Bontsema (1991) and Tchamitchian *et al.* (1993) consider both temperature and CO_2-concentration. Finally, Van Henten (1994) and Tap *et al.* (1996) also incorporated humidity through penalty functions.

4. Characteristics of the models

   There is a wide variety in models used. Since simple models reduce the computational problems related to non-linear optimization, the earlier approaches use small and relatively simple models compared to the ones available in research. For instance, Gal *et al.* (1984) include a model that contains only one state variable which represents the accumulated dry matter during the optimization period. In Seginer and McClendon (1992) the model includes two state variables namely dry

> matter accumulation and leaf area expansion. In both approaches the climate models are simple, algebraic and static, adding no state variable to the problem. More complex climate models have also been used. Tchamitchian (1993), Tap (1993) and Van Henten (1994) emphasize the physical model of the greenhouse. Van Henten (1994) describes the greenhouse climate with three state variables for CO_2-concentration, temperature and humidity of the greenhouse air. The use of simple models may lead to non-realistic results as shown by Tchamitchian (1993); it may also stimulate undesired behavior of the control devices (bang-bang control). There is agreement that more complex models would improve the results of optimal control, if they can be dealt with computationally[107] (*e.g.* Seginer and Sher, 1993).

The approaches discussed above show a variety of ways to implement optimal control principles in greenhouse climate control. Unfortunately, most studies have presented results based on simulation experiments only. The first experiments using an optimization approach have recently been conducted (Tap, in preparation). Preliminary observations, showing the difficult road from theory to implementation, have been reported by Tap *et al.* (1996). Their approach is based on the two timescale decomposition of Van Henten (1994). For the fast optimization problem a receding horizon of 60 minutes is used and control settings are calculated each minute.

Within the optimization approach, attention for user interaction has, until now, been limited[108]. This is not surprising given the normative nature of the approach itself and the phase in which research is presently in. Seginer (1996) suggests manipulation of co-states by growers. Van Straten and Challa (1995) opt for grower-formulated penalty functions on the climate or crop state variables. Van Henten (1994) considers both.

4.4.2.2 AI approaches

Martin-Clouaire *et al.* (1993a) and Martin-Clouaire *et al.* (1993b) propose a constraint satisfaction approach for daily temperature setpoint determination. Their system SERRISTE has the objective to elevate production performance while saving energy expenditure. SERRISTE calculates for three different periods (*i.e.* day, night, and dawn) of a 24-hour cycle, the heating and ventilation temperature setpoints (in the present implementation humidity setpoints are set to default values). These

---

[107] In every optimization problem that is based on search, computational complexity should somehow be dealt with. In many cases it will mean a redefinition of the original problem (*e.g.* Aarts and Lenstra, 1997).

[108] The early publications do not discuss user interaction, however, since the initial ideas have evolved considerably, some kind of grower interaction has been suggested recently (Van Straten and Challa, 1995).

device-related temperature setpoints are determined according to a set of implicit objectives such as: *i*) avoiding problems (*e.g.* Botrytis), *ii*) proper growth and development (vegetative - generative balance) of the crop, and, *iii*) minimizing energy expenditure. Some objectives can be specified by the grower; he can set the mean 24-hour temperature and he can supply a value for a change of the "vigor"-attribute of the crop. The knowledge base in the SERRISTE system is a combination of scientific and practical know-how and is represented as a network of constraint relations. The system explicitly uses the response of crops to average temperatures (*e.g.* De Koning, 1988). Besides above objectives, the system also uses the weather forecast (and corresponding historical values) as input. A prototype of SERRISTE has been implemented on a PC and tested during a five month's trial in a large greenhouse compartment ($\approx$ 300 m^2) at a research station. The system performed comparably to a control treatment (carried out by an extension specialist using a conventional system) but used 10% less energy. Additional trials have been planned but results have not yet been reported.

4.4.2.3 Combined approaches

Ehler and Karlsen (1993) outline the system OPTICO, which is dedicated to CO_2-concentration optimization. It uses a small set of rules and an objective function in which the ventilation rate and the photosynthesis production are important variables. The system determines: the injection scheduling, and the radiation and ventilation dependent CO_2-concentration setpoint levels. Boundaries on the CO_2-concentration levels, maximum CO_2-expenditure and other constraints can be set by the user. The user is also responsible for determining other climate settings (*e.g.* temperature). The system, which has been tested with a sweet pepper crop, increased the number of fruits but reduced the average fruit size. Overall performance reduces the CO_2-input and suggests a fresh weight increase. This system can be seen as an extension to presently available control systems and may easily be put into practice[109].

Gauthier (1992; 1993) reports a generic greenhouse management platform GX. GX is a greenhouse climate management *shell* designed to support knowledge-based control of the greenhouse environment and permitting the implementation of control strategies that dynamically optimize the greenhouse environment. GX has been interfaced with two of the currently available control systems[110]. It supports the

---

[109] The Hoogendoorn company markets a similar system called CARBONAUT. This system calculates an optimal schedule for filling the heat storage tank (during filling, CO_2-dosing can be applied).

[110] Interfacing with currently available systems is difficult since the manufactures do not provide facilities for it, that is, application programming interfaces (API's) do not exist. Achieving proper communication (*i.e.* exchange of settings and measured data) with a Priva system (Priva Agro BV, De Lier, The Netherlands) required a lot of "hacking" (Gauthier, personal communication).

specification and deployment of dynamic strategies defined as setpoint adjustments based on context sensitive information such as: *i*) operator's preference values for certain parameters within the strategy, *ii*) outside climatic conditions, *iii*) crop value, *iv*) energy costs, *v*) weather forecasts. The decision making capabilities of GX are essentially embedded in heuristic rules (*i.e.* "scripts") that convey directly applicable procedural knowledge telling what to do in a given context. Within these rules, "intelligent agents" (*e.g.* agents that can use simulation models) can be directed to perform their designated operation. Because GX is an open (object-oriented) environment it can be extended: agents can be added, new scenario's can be designed using the features already present in the system, etc.. This attribute should make GX very attractive for manufactures of currently available greenhouse control systems, especially because the object-oriented implementation may also offer an opportunity for a gradual shift to more model-based control. The architecture of GX has been tested with success in an industrial size experiment in which lettuce was grown year round; the part of the knowledge base actually concerned with strategies was quite small in this experiment. Although GX system offers a clean and powerful representation framework for a greenhouse production complex, it seems that a thorough evaluation of its ability to express sophisticated management strategies and to permit effective application of them has still to be done.

4.5 Concluding remarks regarding supporting or taking over the decision making activities of the grower

4.5.1 Use of knowledge

The systems described in this chapter are either based on procedural knowledge, on declarative knowledge, or on a combination of both. As discussed in chapter three, the type of knowledge determines its possible use.

Potentially, declarative models allow for planning and determination of optimal management practices. Optimal control systems that consider the climate management task as the problem of finding a sequence of actions that optimize a numerical performance index belong to the class of systems that execute a form of planning. However, this approach has the disadvantage that it needs all relevant knowledge which is of course very difficult.

Approaches based on procedural knowledge address the management task as the problem of finding the appropriate reaction to a given situation by choosing from a set of pre-defined procedures that tell how to control things. The essential task of the management system is then to choose what decision procedure should run and when, and, when several are applicable at the same time, to ensure that they are properly coordinated. Rule-based systems are developed along this line. Because this approach is usually short-sighted, theoretical optimality cannot be reached. Nevertheless, this approach has the advantage of being closer to the working practices of the grower and thus facilitates the representation of his procedural knowledge.

The actual domain knowledge contained in the systems discussed, was limited. Most approaches emphasize certain key variables or processes (and ignore others that are also of importance). Most prominently present are the photosynthesis process (predominantly influenced through the CO_2-concentration) and the temperature variable.

4.5.2 Decision making

Almost all systems discussed in section 4.4 contain objectives that either aim at optimizing photosynthesis, reducing energy consumption or a combination of both. In the latter case competing objectives are expressed in financial units such that they may be compared. This approach may be theoretically attractive, however, it does hold a weakness since not all objectives/variables can be expressed in financial units[111].

It can be observed that some approaches value certain variables or even complete sets of activities of the grower on the basis of their contribution to operational *costs* of the grower (this is especially true for the cost for heating, *e.g.* Tchamitchian *et al.*, 1993) and not on their contribution to the yield or product quality. This may be the reason that most approaches consider climate management instead of nutrient and irrigation management. Likewise, within climate management humidity control is much less emphasized than temperature setpoint determination[112].

A further distinction is that some approaches only consider the short term crop responses and do not take long term effects into account (*e.g.* Hashimoto *et al.*, 1981). Others (*e.g.* Seginer, 1991) consider only (part of) the long term crop responses and ignore short term effects. It may be concluded that ignoring important processes seriously reduces the value of the output of these systems. The notion of optimality that these proposals promise may thus become a hollow phrase. Embedding the optimization process within an overall approach that checks and corrects the optimization results before implementing them, is therefore necessary.

The more recent computer-based approaches recognize that operational greenhouse production management or even climate management as one of its sub-tasks should

---

[111] Zeleny (1974, in: Keen and Scott Morton, 1978) argues that the traditional concepts of optimization, requiring prior definition of a (simple) objective function, are inapplicable in situations involving multiple, conflicting, and incomparable objectives, thus leading to multi-criteria decision making.

[112] Another reason for such a choice may also be our limited knowledge regarding humidity effects on yield or product quality and the occurrence of diseases. This of course seriously affects the possibility to even try to optimize a task like quality management.

not be fully automated and must allow for some kind of user interaction (*e.g.* Van Straten and Challa, 1995). These approaches indicate a number of possibilities to allow the grower to influence the settings of the actuators, for instance: (qualitative) values for certain crop-states (Martin-Clouaire *et al.*, 1993a), penalty functions on certain variables (*e.g.* Van Henten, 1994) or values for co-state variables (Seginer, 1996).

It is argued in chapter two that objectives should not be hardwired into computerized support systems unless they are part of the set of programmed decisions of the grower. Since setpoints or control actions are determined on the basis of these objectives, they cannot be considered programmed decisions unless they are carefully embedded in the *other* objectives of the grower. The questions then become: *i)* is the user interface of these approaches versatile enough to allow the grower to enter his other objectives and, *ii)* has a reliable method been found to integrate the input of the grower with the hardwired objectives? Unfortunately, there has not been much interest in the actual application conditions or the embedding of the suggested approaches[113]. The validation experiments reported consider the performance of the approach either in a research setting (Martin-Clouaire *et al.*, 1993a; Tap *et al.*, 1996) or consider the architecture only (Gauthier, 1992). Therefore, at this point in time, above questions cannot yet be answered.

4.5.3 Evaluation

The systems reviewed in section 4.4 may be characterized as 'use cases', that is, they may be seen as attempts that extend the boundaries of what may be technically achievable given the methods developed in the fields of control theory and/or artificial intelligence (using knowledge available in the horticultural domain). To explore the applicability of a problem solving method, it is generally necessary to change the problem specification to fit the method. In doing so assumptions about and simplifications of the problem are usually needed. How far one may realistically go with these simplifications depends of course on the problem at hand and requires careful study of the problem (and the problem owners). After initial application of the method on a simplified problem in an artificial environment, efforts should be directed to extend the problem solving method to correspond to the actual application conditions.

Unfortunately, none or only limited validation of every approach with respect to situations comparable to that of modern commercial growers has been reported. As a result, the real value of the systems discussed remains unknown.

---

[113] Since (to the author's knowledge) chapter two can be considered the first descriptive analysis of operational decision making in greenhouse vegetable production.

For this kind of application-oriented research a validation methodology is needed[114], however, such a procedure has not yet been (and might never be) formally developed. As in all information systems, an evaluation method for the above approaches should have strong sociological and psychological components since user acceptance is crucial for the success of the above types of systems. In a large part, owing to the traditions in the research community from which these systems have emerged, attention for user involvement has been insufficient. The discussion on evaluation aspects will be continued later since attention for the user's way of working and the influence hereof on what may be seen as ***appropriate computerized support*** is one of the focal points of this thesis.

---

[114] Unless one is only interested in normative comparisons of different scenario's (*e.g.* Leutscher, 1995) the intended user should be involved (and not only during the evaluation).

5. EPILOGUE OF PART I:
THE QUEST FOR APPROPRIATE COMPUTERIZED SUPPORT

5.1 Introduction

The previous three chapters discussed various aspects that are considered important for developing a new design of a computerized support system for operational crop production management.

In chapter two, the characteristics of operational crop production management were discussed. The role of the grower and the presently available control systems were analyzed and recommendations regarding future computerized support were given.

In chapter three the attributes and the availability of knowledge in the domain of crop production were discussed. That chapter provided insight in how knowledge in this domain can be put to use.

Finally, in chapter four, previous attempts for computerized support in operational crop production management were discussed.

It is argued that above three sources of information are indispensable for the design of a new computerized support system for operational crop production management. First of all because one must have a good understanding of the tasks and activities of the grower before one is able to find suitable means to support these tasks and activities. Second, because developing a knowledge based system will not likely be successful without sufficient insight in the attributes and the availability of the knowledge in the domain. And finally, because insight in the previous attempts for additional computerized support will generate information about the implications and advantages/disadvantages of the proposed techniques.

In part two of this thesis, the results of the analysis will be used as a base for a design of a new computerized support system, but first we will discuss the foundation of this design.

5.2 Optimal operational crop production management, a grower's perspective

Computerized support for operational crop production management consists of two main functions. First, it supports the grower during interaction with his computer systems (at present these are the climate control system and the irrigation and nutrition control system), and second, it provides continuous autonomous control of the environment of the crop by these systems. These two functions are distinct although there exist strong interactions between them.

5.2.1 Decision support

Decision support during interaction consists of: *i*) providing information about the past and present state of the greenhouse - crop system and *ii*) assisting in the determination of input for the autonomous control phase. Without underestimating the importance of first function, here, the second function will be emphasized.

In this study computerized support for the grower is the focal point, consequently, the central question is: "How may the grower best be assisted in the determination of input for the autonomous control phase?". Contrary to most of the proposals discussed in section 4.4, the question is not: "How can some kind of (grower-independent) setpoint or control action generating scheme be formulated and embedded[115]?"

In chapter two it was shown that the definition of optimality should be the sole territory of the grower, that is, *he* should determine what must be considered to be optimal during the autonomous control phase. Not only because this is his responsibility as an entrepreneur but also because our insight in what may 'objectively' be considered optimal crop production is not yet satisfactory. The seven main aspects (amount and quality of the product, timing of production, maintaining production potential, reduction of costs and production risks, and compliance to socio-economic constraints) are too high level, incommensurable and cannot be given a value. Moreover, individual growers will have different opinions about optimality whereas the conditions under which they produce vary widely. Finally, it is argued that presently available domain models are not sufficiently accurate (nor complete), and are computationally inappropriate[116] in the light of the suggested approaches (section 4.4).

5.2.2 Input of the grower

During the interaction phase the grower has to decide about what to enter into the system, that is, what particular values to assign to the decision variables[117]. What

---

[115] The proposals discussed in section 4.4 primarily focused on formulation of a control action generation scheme; in section 4.5 it was pointed out that embedding such scheme in an overall framework did not get sufficient attention.

[116] These two characteristics are to some extent antagonistic. Some of the more accurate models contain many state variables which makes their use computationally difficult.

[117] The 'language' of the control system consists of a set of variables that are present in its interface. The *interface* variables are defined as the variables that represent meaningful concepts for the grower within the context of the control system. The

Continued →

he will enter depends in part on what he wants to pursue, that is, his objectives. The grower's objectives comprise desired state trajectories of his crop and its environment, and closely related to these, his decisions represent his commitment to reach and pass through particular desirable states.

The grower effectuates his commitment by assigning values to the available decision variables. Which decision variables the grower can use, depends on the computerized support system itself. In chapter three it was explained that the interface variables are situated at three levels, namely:

- the crop level, this level contains variables that represent characteristics of the crop,
- the environment level, this level contains variables that represent attributes of the root and shoot environment, and
- the control level, this level contains variables that are related to the set of control devices.

It was concluded that the objectives of the grower are mostly situated at the crop and environment level, while the presently available control systems only contain decision variables that belong to the environment and control level. This 'gap' between the grower's objectives and the - matching - assignments to the available decision variables requires the grower to carry out a difficult and knowledge intensive translation process. It is precisely this translation process that will be explicitly supported by the computerized support system described in the remainder of this thesis.

5.2.3 The role of domain knowledge

It has been suggested in chapter two that the grower can be assisted in this translation process by replacing some of his heuristics by computerized versions (*i.e.* models). The principal idea behind this type of decision support is to *allow* (and not to force) the grower to enter his objectives at the crop level. Domain knowledge that fits into such an approach (*i.e.* domain knowledge that does not contain implicit objectives) may then be used to infer inputs for the autonomous control phase.

Whether entering decisions at the crop level is possible, depends in the first place upon the availability of models for this deduction process. Furthermore, the support system must contain an inference mechanism that carries out this deduction process by itself. This mechanism combines the input (*i.e.* settings or constraints) of the grower entered at the crop, environment and control levels into a balanced set of inputs for the autonomous control phase. For the grower, the advantages of this type of decision support may be that:

---

decision variables are the subset of the interface variables to which the grower assigns values.

- the approach allows explicit control of crop states for which appropriate models are available, hence more explicit and direct intervention with regard to his objectives is possible,
- objectives and input partially reside at the same level, the grower determines what must be pursued, the decision support system partly determines how[118],
- the system contains heuristics/models which may be better than the grower's own, and
- through these models, the system allows the grower to learn about crop processes.

A possible drawback of such approach is that the grower must now deliberate about his complete set of objectives and decide about some of them individually. Additionally, he must decide at multiple levels (*i.e.* the crop, environment and control level). It is not yet clear whether a grower may become comfortable with such an approach.

5.2.4 *The autonomous control phase*

Focusing on the assistance during the input generation process does not imply that the autonomous control phase itself may not be improved. It merely suggests that the only constraint for such improvement is that the control actions should reflect the decisions made by the grower. That is, when the methods applied during the autonomous control phase carry out some kind of optimization, this must be in agreement with the input of the grower. Optimization within the boundaries of the pre-processed (by the inference mechanism) input of the grower is a possibility. It could be advantageous in the sense that it may lead to efficient implementation (*e.g.* regarding the use of energy) of the grower's decisions.

5.3 Implications of a support system that is based on delegation

The above approach, like almost every approach suggested in the reviewed literature (section 4.4), implies a shift from action- and boundary-oriented settings (used in the presently available control systems, see section 2.4.6) to state-oriented settings, that is, settings that involve state variables and desired values. Within this

---

[118] Of course, at first the grower will be very cautious about the behavior of the system. This type of delegation is in close analogy with the decision structure in organizations with multiple decision layers (*i.e.* executives → middle management → shop floor). Decisions at the highest level are passed on to lower levels that reformulate them and make them more explicit and may pass them on to even lower levels. At the bottom level these translated decisions are carried out in practice.

approach the grower states *what* the system should achieve and not (or only partly) *how* it should behave.

When introducing such a concept in which state-oriented settings play a dominant role one should recognize at least two major differences with action- and boundary-oriented settings.

Firstly, the settings represent states that are intended to be reached, instead of actions to carry out (triggered by certain conditions). The feasibility of a combination of settings can now become a problem and depends on: *i*) the values assigned to the variables, *ii*) the outside disturbances and *iii*) the initial conditions. Since attainability problems will occur (*i.e.* most likely under the more extreme weather conditions), some kind of setting integration scheme[119] should be available (*e.g.* methods that assign priorities to individual settings and/or constraints). In present control systems the issue of attainability does not play a role because action-oriented settings can always be carried out and boundary-oriented settings are only set on variables that represent the actuators.

Secondly, although this issue has received little attention in the proposals discussed in the previous chapter, the use of state-oriented settings requires quite another view on his settings from the grower. This new type of settings should be seen as *direct* mappings of the grower's objectives as opposed to the settings in presently available control systems of which the *overall behavior* represents his total set of objectives. In a new approach in which state-oriented input is dominant, the grower must be familiar with the way a support system carries out his preferences.

---

[119] In the optimal control approaches (section 4.4.2.1) this issue is automatically solved through the so-called profit production rate which is being optimized. However, this approach has other more fundamental drawbacks *e.g.* formulating incommensurable quantities in financial units.

PART II

Design

INTRODUCTION TO THE DESIGN PART

Chapter five concluded the analysis of the various aspects of problem domain. The analysis has resulted in a clear focus on the decision task to be supported by the new supervisory control system.

The term 'supervisory control system' is preferred over decision support system, since the system contains both a control function as well as a decision support function (*i.e.* it can be seen as a control system of a higher order). The foundation of this system is: *i*) the introduction of domain knowledge into the system and the enrichment of the system's interface with decision variables at the crop level, thus allowing for more direct and explicit control, *ii*) the availability of an inference mechanism that is able to process the domain knowledge and related decision variables, and *iii*) the possibility to learn about decision variables at the crop level, and the flexibility to gradually use them more.

In chapter six to nine these basic attributes will be worked out in much more detail. In chapter six the requirements of this system will be discussed. Within the total set of requirements, special emphasis will be placed upon the requirements that relate to both user interaction and the decision task to be supported. This chapter also contains the global functional design of the supervisory control system. This design points out the functional relationships between the decision support part and the control part of the overall system.

The most interesting part of the supervisory control system is the subject of chapter seven. In this chapter the inference engine is worked out in detail. It will be shown that constraint reasoning has particular features that makes it an attractive technology to carry out this specific inference task.

Afterwards, in chapter eight, the workings of the inference engine will be shown for a number of examples.

Chapter nine concludes the second part of this thesis and evaluates the overall system design.

6. REQUIREMENTS SPECIFICATION AND GLOBAL FUNCTIONAL DESIGN

6.1 Introduction

This chapter presents the requirements specification and the global functional design of a supervisory control system that supports the grower's operational crop production management task. The requirements specification[120] outlines the entire system although special emphasis will be directed towards the decision support task of the overall system.

Whenever requirements are specified its level of detail must be specified. In commercial software development projects[121] this choice depends amongst others on the kind of system that needs to be developed[122], on the project organization that develops the system, and on the amount of time that can be invested in the specification. Here, attention will be focused on the functional attributes of the system, and, given the scientific nature of this work, other more technical requirements (*e.g.* user interface attributes, database interaction and the like) will receive little attention.

Furthermore, the requirements and the global functional design are formulated in such a detail that they sufficiently show that the main ideas behind the approach can indeed be embedded in a supervisory control system that supports the grower in his operational production management task. Our description portrays an overall system concept of which some crucial parts will be worked out in more detail in later chapters. It should be stressed that the requirements specification and global functional design are for a large part concept-oriented since producing a complete working system is not pursued within the scope of this thesis.

The remainder of this chapter is structured as follows. First, a general description of the overall system will be given. Secondly, on the basis of this description, the func-

---

[120] In this chapter the IEEE Standard 830 for software requirements specification. (IEEE (1984) in Van Vliet, 1993) will be roughly followed.

[121] In commercial software development projects the requirements specification is usually part of the contract between the customer and the developer (Van Vliet, 1993).

[122] It is often stated (*e.g.* Van Vliet, 1993) that a requirements specification must - among others - be complete and unambiguous. In practice, these strict conditions will only be realized in very specific cases. In software projects where sophisticated decision support systems are being created, the extent of the ultimate amount of decision support cannot accurately be determined beforehand. In such cases an evolutionary development approach can be followed, where requirements are defined and updated during the course of the project.

tional requirements will be worked out in detail. Finally, a global design will be given that may accomplish the tasks set out in the requirements.

6.2 General description

6.2.1 System perspective

6.2.1.1 Scope of the supervisory control system

The supervisory control system concept is being developed based on the idea that more effective and efficient crop production management should be possible when we succeed in making better use of the available knowledge on crop and environmental processes (*e.g.* Challa *et al.*, 1988). The general idea is not new, however, chapter four reveals that realizing this idea and creating a workable tool for the grower is not a straightforward undertaking. The specific approach taken here differs from earlier attempts because it has been based on careful analysis of the problem of the grower.

The supervisory control system has to carry out it tasks during two phases. First, it should support the grower while interacting with his system. During this phase the grower gathers information from the system, makes decisions, and adjusts settings. Second, during the autonomous control phase, the supervisory control system manages the environment of the crop and tries to adhere to the settings entered by the grower. Consequently, the system has three mayor functions:

- decision support by providing the grower with information of the crop and its environment,
- interaction with the grower, and
- decision making for controlling shoot and root environment.

Hence, the system should have decision making as well as decision supporting functions.

6.2.1.2 Strategic solution direction of the supervisory control system

The previous chapter discussed the strategic solution direction of the supervisory control system. It was concluded that a system in which a grower can enter his settings at multiple levels can be a valuable contribution because - amongst others - it partly relieves the grower from the difficult translation process from objectives to settings. The key contribution of such a system is allowing the grower to delegate some of his decision making to the supervisory control system, however it should do so without reducing the grower's span of control (unless with his consent).

Designing a system in which a grower can enter his settings at multiple levels has important consequences for the functions mentioned above. W.r.t. the decision support function for providing the grower with information, the system must, for instance, provide reference values for the specific crop-related settings. Regarding its interactivity, the system's user interface must be sufficiently versatile that the grower is able to enter his settings (in a manner that suits him). Given the possibility that not all settings can be realized, his settings necessarily contain some kind of preference distribution. Finally, w.r.t. the decision making function for controlling shoot and root environment, the system must include an inference mechanism. This mechanism must, for instance, be able to deal with (various types of) crop models, it must be able to combine the settings of the grower with information on the current crop and environmental state, and it has to resolve conflicts on the basis of the preferences set by the grower.

These rather abstract requirements will be worked out in more detail further on.

6.2.1.3 The supervisory control system in its context

The supervisory control system (Figure 6-1) will mainly operate as a stand-alone system. It computes control actions from: *i*) settings entered by the grower, *ii*) measurements obtained from the greenhouse and, *iii*) information acquired from external databases. Communication between the supervisory control system and its measurement and control devices is frequent (in the order of seconds to minutes), while interaction with the grower and communication with external databases only occurs at most a few times a day.

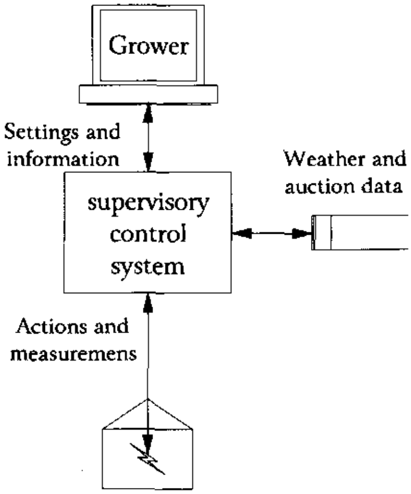


Figure 6-1 The supervisory control system in its context.

6.2.1.4 Implicit requirements

One of the fundamental requirements of the supervisory control system is its embedding in the working habits of the grower. The supervisory control system should support rather than replace the grower, therefore, it should not take decisions that are considered the privilege of grower. It should support his decision making and should therefore be suited to the model of the decision making process (section 2.5.2) and to the grower's integrated view with respect to the four tasks areas (Figure 2-3).

Since working practices change and are supposed to change, the supervisory control system must be a flexible tool and should offer the grower a path from his present to his future control practices (*e.g.* an adaptable user interface in which the interaction style can be adjusted). It is believed that the introduction of a new system should not be a paradigm shift for the grower; the supervisory control system should therefore use more or less the same 'language', that is, interface variables, with which the grower is presently familiar.

The supervisory control system should also support the grower in learning more about the control mechanisms and the processes/phenomena in the crop.

6.2.2 The functions of the supervisory control system

The following sub-sections first describe the functions of the supervisory control system in general terms, later these functions will be described in more detail.

6.2.2.1 Provide information

The supervisory control system should provide the grower with information about his crop and its environment. This information is provided though sensors and simulation models. The supervisory control system should also obtain information from external sources like extension service agencies, the auction, and meteorological agencies. Together with his direct observations of the crop and the internal and external environment, the above pieces of information act as a source of reference during the grower's decision making process.

To be able to make informed decisions at the crop level the grower must have reference values for variables at this level (examples of such variables are: transpiration rate, assimilation rate, fruit ripening rate). The system should therefore provide these reference values. At present the grower has very limited experience with variables describing crop process rates. Acquiring insight in what proper values for such variables (in his particular situation) may be, is required before he can use these variables as a basis for generating control decisions during the autonomous control phase.

6.2.2.2 Determining and entering settings

During his decision making process, the grower will consider (a) specific objective(s) to work on[123]. Such an objective may imply a change in the rate of some crop proc-

---

[123] As already stated (section 2.5) such an objective is not seen in isolation but is merely highlighted from the complete set of objectives.

ess (*e.g.* the ripening rate of the tomatoes) that may best be achieved through changes in the settings of the control systems (as opposed to manual interventions[124]). The grower must then decide about or translate the objective at the crop level into setting changes - if any - that can be entered in the supervisory control system. To assist the grower in this task, the supervisory control system allows - where possible - the grower to enter his settings at the crop level and performs this translation process for him by using models of crop processes. Hence, interaction at the crop level could reduce the complexity of the grower's decision making process[125]. Unfortunately, not all crop and environmental processes have been modelled with sufficient accuracy to allow for this kind of automated translation, thus requiring the grower to enter settings at the environment and control level.

6.2.2.3 Scenario analysis

As discussed in section 5.3 the settings of the grower in the supervisory control system have a different meaning (and consequences) as compared to the settings he can enter in his present control systems. Therefore the supervisory control system should allow the grower to foresee the results of his decisions by simulating the future values of variables he considers important. This analysis can serve two purposes, first the grower can pinpoint aberrant decisions, or the incompatibility between two or more decisions (given initial states and expected weather conditions). Second, the grower can simulate the system's behavior under various weather conditions in order to assess the possible courses of development (and thus play what-if analyses).

6.2.2.4 Autonomous control

The combination of the settings entered by the grower must eventually result in actuator values. The supervisory control system must therefore be able to integrate the settings entered at various levels and calculate appropriate actuator values by taking into account: *i)* measured and simulated values of the current state of the greenhouse-crop system, and *ii)* predictions about future states. During this process a conflict resolution mechanism should be available to deal with the preferences that have been entered by the grower.

A setting at the crop level may have its effect on the values of actuators in both the root environment and the shoot environment. The supervisory control system manages therefore the climate control sub-system as well as the irrigation and nutrition control sub-system.

---

[124] It might be possible to develop a system that recommends on manual operations to the grower, however such a system will not be pursued within the scope of this work.

[125] In certain situations a setting at the crop level could replace a series of settings at the climate level.

It is important that the actuator values can be calculated and implemented in due time to enable the supervisory control system to respond adequately to the fast disturbances on the system (*i.e.* mainly changes in the light intensity and rain).

The ability to integrate the decisions entered at various levels properly will be the central contribution of this system since it is a prerequisite for allowing the grower to enter some of his decisions at the crop level.

6.2.3 User characteristics

Growers are generally the sole users of the supervisory control system. They are highly knowledgeable with the task at hand. The supervisory control system should be flexible in that it offers different interaction possibilities to accommodate various types of grower (sections 2.3.1 and 2.5.1). This flexibility may be realized by offering a variety of interface variables and levels of use.

Growers have an integrated view on crop production in that they consider the possibilities to influence the growth and development of their crop in relation to each other (section 2.2.4.). The supervisory control system should support this view. This can be achieved by carefully integrating climate and irrigation–nutrition management support functions into one support system.

6.2.4 General constraints

The supervisory control system should be sufficiently general so that it will be able to operate on a range of crops and in various greenhouse configurations. The system should contain crop specific knowledge since this is believed to be a critical success factor in further progress in the operation management of greenhouse crops (*e.g.* Challa *et al.*, 1988). Since the system will contain crop specific knowledge, it will model a part of the reality that can be observed in the greenhouse. To ensure that the internal state representation sufficiently mirrors its real counterpart, parameter identification procedures are needed. The requirements concerning system-generality and the inclusion of crop specific knowledge, demands that the domain knowledge can be easily 'plugged in' and/or customized to other operating environments.

The supervisory control system should not take decisions that may be considered judgmental without explicit authorization of the grower (section 2.3.2.2). Since only the grower himself can determine what can be considered non-judgemental, the system should offer means to configure the set of choices left under the responsibility of the system.

The decisions entered by the grower will in part be state-based, therefore, the supervisory control system should accommodate a conflict resolution procedure for the autonomous control phase. This procedure should be able to use some kind of preference ordering mechanism in which the grower can prioritize his decisions.

This simply means that there is some flexibility in the desirable states and that some states are preferred over others (*i.e.* their preferences depend amongst others on the actual weather conditions).

Like any other software system, the supervisory control system is likely to require improvement and adjustment. Apart from the improvement and/or replacement of the models in the knowledge base, adjustment of the software to new control equipment should be possible. These demands require a high level of modularization, especially of the knowledge base.

6.2.5 Assumptions and dependencies

The use of the supervisory control system requires that the grower may become comfortable with: *i*) decision making at multiple levels (*i.e.* the control, environment and crop level), *ii*) prioritizing his decisions by way of preference distributions. The latter is considered necessary because during the autonomous control phase it will often not be possible to realize all the grower's objectives in full. With respect to the first, the grower is not required to use all levels, however if he does, he must understand their roles and relationships.

6.3 Functional requirements

This section elaborates on the system's main functions discussed in section 6.2.2.

6.3.1 Provide information

6.3.1.1 Past and present states of the greenhouse - crop system

The supervisory control system should offer the grower extensive possibilities to display (and hardcopy) information about the state of the greenhouse - crop system and its outside environment. The grower should be able to display the values of all the interface variables[126] in appropriate formats. The system should allow the grower to average and cumulate variables over time and over space (*i.e.* per m^2, per plant). The grower should be able to compare present and historical (*e.g.* last year) values and display them in appropriate graphical formats. On the whole, the system

---

[126] The 'language' of the supervisory control system consists of a set of variables that are present in its interface. The *interface* variables are defined as the variables that represent meaningful concepts for the grower within the context of the supervisory control system. The *decision* variables are the subset of the interface variables which the grower can constrain. The *control* variables are the subset of the decision variables that directly relate to the available control devices and which values are transmitted to the control systems at regular intervals.

should offer many of the functions that are present in modern on-line analysis and processing (OLAP) tools.

To provide the above information, the supervisory control system should necessarily include a database to store its data.

6.3.1.2 External sources

Like the above, the supervisory control system should provide the grower with up-to-date information about expected weather and product prices. The system should also be able to collect information from extension agencies. Additionally, it should offer provisions to allow the grower's personal advisor to remotely collect information about the state of the greenhouse - crop system and remotely supply the grower with advice without having to visit the nursery.

6.3.2 Entering settings by the grower

A setting or *input constraint* is a restriction[127] placed on the domain of a decision variable by the grower. In this way the grower directs the supervisory control system to maintain the value of the variable within the given bounds. The term 'constraint' can be seen as a generalization of the term 'setpoint' (the latter denotes the value that currently available control systems strive for).

Input constraints may also be stated qualitatively if the system accommodates qualitative decision variables. The use of these decision variables may be advantageous if no quantitative counterparts are available, which is the case for a number of important crop processes (section 3.7.2.). Eventually, these qualitatively stated input constraints must play a role in the determination of quantitative control values.

Input constraints are flexible specifications of acceptable choices and are given a preference ordering over a set of non-excluded alternatives. The preference ordering is advantageous and necessary during the autonomous control phase because it can be used to realize robustness in the behavior of the system. The preference ordering allows the system to work autonomously and still comply with the decisions of the grower[128].

---

[127] A constraint reduces the domain of a variable to a smaller - possibly empty - interval (in case of continuous domains) or to a smaller number of values (in case of discrete domains).

[128] The grower's input constraints are mainly related to state or rate variables that may not always be realized due to unfavorable external conditions (*e.g.* low light intensity). During the autonomous control phase a preference ordering mechanism can be used to temporarily switch to a lower preference level without the need for the grower to interfere and adjust his settings. Hence, a preference order can help the

Continued →

The level of preference and the width of the preferred interval of an input constraint are linked as can be seen in Figure 6-2. The width of the input constraint decreases with an increase in the level of preference (*i.e.* the higher the preference requirement, the smaller the set of satisfactory values). The shape of the preference distribution in Figure 6-2 indicates that the preference ordering is not necessarily symmetrical. The preference distribution should be seen as 'shorthand' for a set of constraints with varying preferences on the same decision variable.

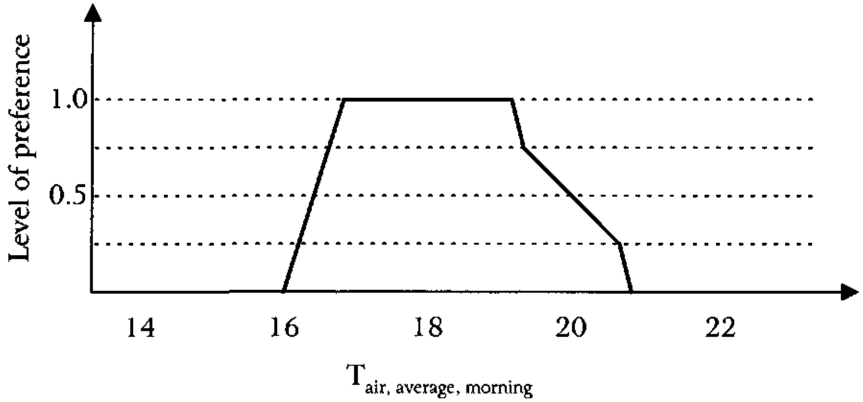


Figure 6-2 Example of a preference distribution on a variable

The grower should also be able to state a constraint conditionally (*i.e.* the constraint only applies when the condition is met). Furthermore, input constraints can be time-dependent, that is, only apply during a certain period.

While entering setpoints in currently available systems, growers generally identify the following four periods[129] within the 24-hour cycle: night, after night, day, pre-night. Over these periods a grower will usually have different settings for the same variable. In the new supervisory control system the grower may identify additional periods (*e.g.* Figure 6-3). More importantly, in addition to the 'momentaneous' state variables, the grower should be able to constrain their integral and average values[130].

---

grower to profit from the opportunities of the external conditions (light) in the best possible way without the need for frequent manual adjustment.

[129] The exact width, and the start and end points of these periods vary over the season. They may also vary from grower to grower. The phases 'pre-night' and 'after night' refer approximately to dusk, respectively dawn; these phases are used by growers.

[130] *I.e.* constrain the *average* or *cumulative* value for a particular period such that it stays between certain bounds, for instance: $\text{Transpiration}_{\text{cumulative,afternoon}} \in [0.5, 0.6] \ell \cdot m^{-2}$ or $T_{\text{air, average, after-night}} \in [18,19]$ degrees C.

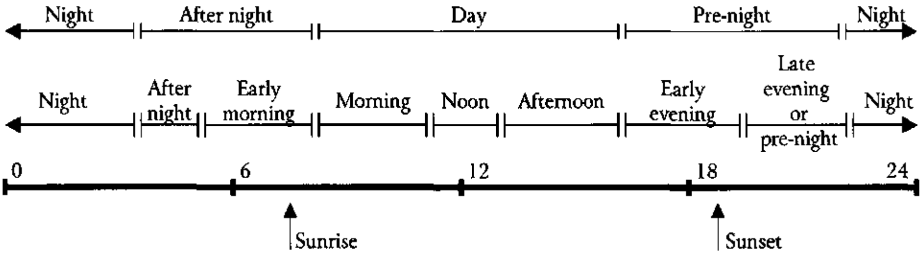


Figure 6-3 Examples of periods within a 24 hour cycle.

Finally, the grower should be able to use 'of the shelf' sets of constraint relations, that can be plugged into the system, to tackle and/or prevent a particular problem.

6.3.3 Determining control actions on the basis of input of the grower

Interaction of the grower with the supervisory control system will ultimately result in a set of input constraints that are used during the autonomous control phase. In this phase the supervisory control system must continuously (*i.e.* approximately every minute) generate values for all available control devices. Table 6-1 shows the control devices commonly used in a tomato production[131].

The supervisory control system should contain (a) suitable inference and representation mechanism(s) to carry out this task. This mechanism should be able to:

- deal with the preference distribution on the input constraints entered by the grower,
- process time dependent and conditional input constraints,
- accommodate uncertainty and imprecision issues (*e.g.* regarding weather), and
- use the variety of available domain models (*i.e.* both qualitative and quantitative).

Above requirements are considered to be strong. Especially because of the need to frequently generate control actions, and the fact that the time constants of important processes show large differences (combining them is computationally difficult),

---

[131] Typically, the control devices are controlled independently, for each compartment in a greenhouse (except for CO_2-input and roof cooling). In this discussion a greenhouse is assumed to consist of only one compartment with one set of control devices.

a system containing two separate inference mechanisms will be suggested. Both mechanisms will be further explained in section 6.5.

Table 6-1 Control variables in tomato production.

Control variable	type of value change and dimension
Heating pipe temperature	continuous in °C
Temperature growth pipe	continuous in °C
Place mobile growth pipe	continuous vertical place in m
Ventilation window aperture (lee)	continuous in % of maximum
Ventilation window aperture (luff)	continuous in % of maximum
Roof cooling	on/off
CO_2-input timing	on/off in g $CO_2 \cdot s^{-1}$
Energy screen aperture	continuous in % of maximum
pH irrigated	continuous
EC irrigated	continuous in $mS \cdot cm^{-1}$
Tickle turns	on/off

6.4 Additional requirements

6.4.1 Domain models

The models contained in the knowledge base of the supervisory control system should have a proven track record[132] and should describe the processes in the greenhouse–crop system in a sufficiently accurate manner. In the first place, this requires that a model has sufficient predictive capabilities and is appropriate w.r.t. the inference process for which it is used.

Furthermore, all control devices should be present in the models to which they may apply. For instance, a model describing the greenhouse climate will probably not be accepted by the grower when a so-called 'growth pipe' is not included in the model (even though it may be shown that the influence of the growth pipe on the model outcome may be limited).

Moreover, the models should allow for efficient computation, and finally, where possible, automatic parameter identification procedures should be available.

---

[132] Judging models will be a valuable, yet necessary and never ending exercise, that should be carried out by scientists, extension specialists and growers together.

These requirements may imply substantial reformulation of the (research) models presently available[133].

6.4.2 *Parameter identification and calibration*

With respect to parameter identification and calibration, the following requirements hold. In the first place, the fixed attributes of the greenhouse, its location and its control devices should be adequately incorporated in the parameters of the models. Second, the characteristics of the crop should be identified, since growers change their varieties on a regular basis (*e.g.* every other season), values for cultivar-specific parameters in the models should be provided for (*e.g.* by seed firms or extension service agencies). Third, at the beginning of the season the initial crop condition should be described carefully and comprehensively.

Finally, the status of the crop represented in the supervisory control system should be kept up–to–date such that it resembles its real counterpart accurately. Procedures should be available such that the grower, his consultant, or his employees can carry out the necessary measurements and adjust the model parameters accordingly.

With respect to the root environment, the presently applied manual procedures for nutrient analysis of the root environment are likely to remain necessary since ion-specific sensors are not yet widespread.

6.4.3 *System extension and maintenance*

Like any software system, the supervisory control system should be extensible and maintainable. Extension and maintenance refer to improvement of the system's functions and knowledge base.

Improving the knowledge base will be a continuous activity. For the grower it may be advantageous if the knowledge base is modular and complies to a predefined 'open' standard[134]. Such a standard is not yet available and should be developed so that it allows model builders (other than the firm that constructed the supervisory control system) to develop models that can easily be plugged into the knowledge base of the supervisory control system. Components of such standard may be: *i*) an initial description of the entities (+ attributes) in crop and environment; *ii*) procedures that describe how such a description may be extended; *iii*) an initial model base (*i.e.* the behavioral part of the entities in the knowledge base); *iv*) procedures that describe how model relations should be formulated (*i.e.* their interface and

---

[133] *E.g.* models describing the greenhouse climate may require a vertical component so the vertically mobile growth pipe can be included. These models should also be extended to incorporate (energy) screens.

[134] Object-oriented modelling is a promising approach to set up a standard framework.

(side)effects); etc.. In short, such standard comprises the 'application programming interface' (API) of the knowledge base.

6.5 Global functional design

6.5.1 General solution direction

On the basis of the above requirements, a global functional architecture may now be proposed. Within this design, the continuous generation of actuator values based on the settings entered by the grower will be carried out in two phases[135].

The first step will be handled by the inference engine of the supervisory control system while the second step will be carried out by two setting calculation mechanisms (one for climate, and one for irrigation–nutrient control). During the first phase the decisions at the crop, climate and control level will be combined to more detailed 'derived' decisions or *output constraints*. From the latter the actual control actions at the actuator level will be calculated.

The rate of change of the information (*i.e.* crop state and weather predictions) in the first phase is slow enough to have the inference engine run approximately once per hour. The continuous changes in the outside weather conditions require frequent adjustment of the climate control devices. Adjustment of these devices should take place approximately once a minute. The rationale behind this two step inference process is the following: *i*) the timesteps of important process models to be included in the knowledge base show large differences (sections 2.2 and 3.6); combining them is computationally difficult, and *ii*) the inference mechanism uses diverse types of knowledge (*i.e.* qualitative and quantitative, procedural and declarative, and probabilistic uncertainty) that could cause logical problems (*i.e.* noncompliance to the input of the grower at certain preference levels) that have to be resolved in an iterative fashion[136].

---

[135] Hierarchical decomposition have been proposed by other researchers, see for instance Udink ten Cate (1983).

[136] Additionally, a coupling is planned with an optimization procedure that is being developed in a complementary research project (Tap *et al.* 1996; Tap, in preparation).

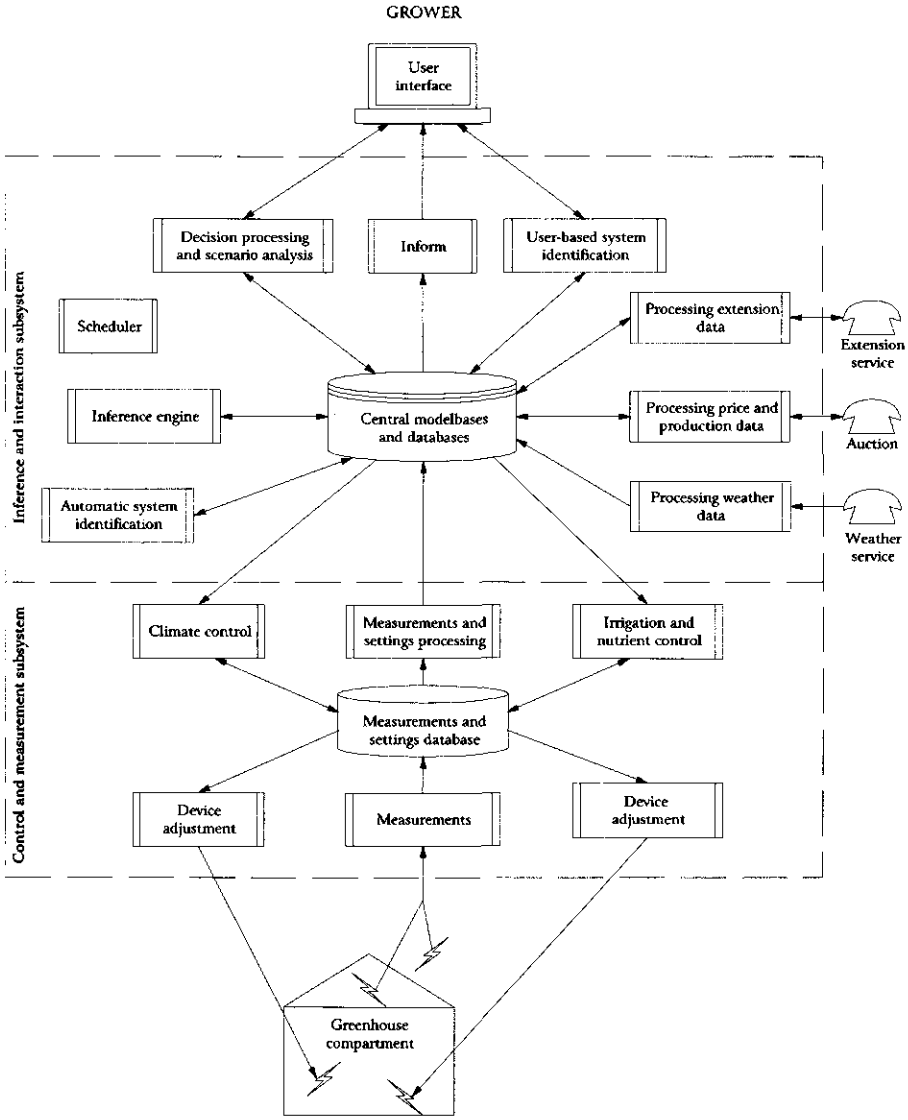


Figure 6-4 Functional architecture of the supervisory control system

The architecture of the supervisory control system shown in Figure 6-4 reflects the above proposition. This figure shows the services or modules within the system and the data flows that connect them. In this design two subsystems can be identified: the interaction and inference subsystem and the control and measurement subsystem. Both sub-systems contain a number of services that bring about their functionality.

Technically, Figure 6-4 shows an architecture in which the individual modules have no apparent interaction. The reasons for a rather loose coupling between the modules of the supervisory control system are related to aspects like: maintainability, up-to-date-ness, and straightforward communication. Every service within the supervisory control system gets its data (and models) from the core model and databases (resp. measurements and settings database). Every information producing service is responsible for writing the information it generates to the respective databases in a timely fashion. In this way the content of the databases is always up-to-date. To keep intra-system communication simple, the individual services do not communicate with each other, they only interact with the system's databases and with the scheduler (in case of the interaction and inference subsystem[137]). The scheduler is central coordinating service within the interaction and inference subsystem. It coordinates the operations of the other services, and activates them when necessary.

6.5.2 The interaction and inference subsystem

In short, the interaction and inference subsystem is responsible for:

- handling the interaction between the grower and the supervisory control system,
- communicating with external databases, and
- generating output for the control action determination mechanisms.

The modules that realize these responsibilities will be discussed individually.

6.5.2.1 Scheduler

The scheduler may be considered the spider in the web of the interaction and inference subsystem. It acts as a mediator[138] between the various modules (which are naturally concurrent). The scheduler sends messages to the different modules thereby instructing them to perform their designated task.

The scheduler 'knows' what the internal states of the interaction and inference subsystem should be and acts on the basis of events generated by other modules. During autonomous control these events mainly come from the inference engine and the control and measurement subsystem. During grower–system interaction the

---

[137] Within the control and measurement subsystem, a separate scheduler module is not anticipated since its operation may be implemented in a cyclic manner in which each service accomplishes its task in a certain predetermined order.

[138] Using a mediator, instead of having the various modules communicate with each other, reduces the complexity of inter-module communication (Gamma *et al.*, 1994).

scheduler obtains additional events from the decision processing and scenario analysis module.

6.5.2.2 Inference engine

The inference engine can be seen as the core reasoning module within the interaction and inference subsystem. The problem solving methods of this module infers output constraints on the basis of the decisions of the grower, the present state of the greenhouse - crop system, and predictions about the future. It uses the model relations that are stored in the central model base.

The inference engine includes a conflict resolution mechanism which uses the preferences entered by the grower. It can reason with both qualitative and quantitative knowledge since the input of the grower as well as the knowledge relations are partly qualitative and partly quantitative. The inference engine will produce a solution (*i.e.* a set of output constraints) with the highest feasible preference level.

6.5.2.3 Central model and databases

The interaction and inference subsystem contains a model and database. The model base stores the domain knowledge and model relations used by the various modules. The database contains the past and present state of the greenhouse–crop system and the settings and preferences entered by the grower.

6.5.2.4 Decision processing and scenario analysis

This module manages the grower–system interaction. It preprocesses the input of the grower such that it can be treated further by the other modules. It feeds the agenda of the scheduler and does some ground work for the inference engine (*e.g.* preparing climate data for running a scenario).

6.5.2.5 Inform

The inform module is responsible for presenting the state of the greenhouse–crop system and other more static attributes stored in the model and databases. The state of the greenhouse–crop system can be visualized in various formats (tables, graphs, etc.).

6.5.2.6 User-based system identification

This module tries to ensure that the representation of the greenhouse–crop system matches its factual counterpart. It updates the parameters of the (crop) models on the basis of measurements carried out by the grower and his staff. On the basis of a schedule and the status of the greenhouse–crop system it determines when the grower needs to supply information.

6.5.2.7 Processing external data

Three modules are responsible for interaction with external sources. One module is responsible for handling the interaction with the grower's personal advisor or other extension agencies. The grower's personal advisor can be authorized to log on remotely and inspect the greenhouse–crop system's status. Another module fetches price and production data from the auction server and sends production data to it. The third server is responsible for obtaining local weather reports (typically two times a day).

6.5.2.8 Automatic system identification

This module uses the measurements from the greenhouse–crop system to automatically adjust parameters of the models used in the inference engine and in the control action determination mechanisms.

6.5.3 The control and measurement subsystem

The control subsystem is responsible for timely adjustment of the control devices of both the climate as well as the irrigation and nutrition control devices. It is also responsible for taking measurements of various attributes in the greenhouse–crop system.

6.5.3.1 Climate control

This module implements the setting calculation mechanism that controls the shoot environment. It uses an optimization procedure based on optimal control theory (Tap, in preparation). Its objective function (and constraints) are determined on the basis of output generated by both the inference engine and the module that processes the decisions of the grower. This module calculates for each timestep (typically one minute), the settings for the control variables.

6.5.3.2 Irrigation and nutrient control

This module implements the setting calculation mechanism that generates control actions for the root environment. It determines the irrigation, pH and EC strategy on the basis of the constraints that are provided by the inference engine. The root environment is not subjected to fast disturbances. Therefore the calculation frequency of the setting calculation mechanism can be less than the one that controls the shoot environment.

6.5.3.3 Measurements

This module implements the measurement system. It frequently polls the measuring devices in the greenhouse–crop system (including the weather station outside the greenhouse). Measurements are stored in the measurements and settings database.

6.5.3.4 Measurements and settings database

This database stores the measurements taken from the greenhouse–crop system, the settings generated by the setting calculation mechanisms and additional information needed in the control and measurement subsystem.

6.5.3.5 Measurement and setting processing

The measurements and settings stored in the database of the control and measurement subsystem are processed (*e.g.* accumulated, averaged, corrected for missing values, etc.) and stored in the central database by this module.

6.5.3.6 Device adjustment

Two modules calculate the final device adjustments (*e.g.* number of seconds a valve needs to be opened/closed, etc.), one for the climate control and one nutrient–irrigation control devices.

6.6 Elaborating on the design

The global functional design shows that the system carries out its primary support task through a three step process. First, the system supports the grower in determining and entering his settings, secondly, these settings are processed into output constraints, and third, the setting calculation mechanisms try to realize these constraints.

In carrying out these tasks the supervisory control system contains, compared to presently available systems, a number of new functions. Many functions of this system such as displaying measured data, contacting a weather agency, etc. will not be scientifically innovative but are necessary for its proper functioning.

Here, only the inference engine of the interaction and inference subsystem will be further emphasized because this module is considered to be its most scientifically innovative part.

6.6.1 Criteria determining the choice of a problem solving method

Selection of a problem solving method for the inference engine requires examining both knowledge representation as well as inference issues since they are closely related (Rich and Knight, 1991). In this work the selection of such a method has initially been representation driven, since utilizing the available numerical models was considered important. As it was recognized that qualitative knowledge is needed to supplement the shortcomings of the numerical models, the representation framework needs to be capable of representing both qualitative and quantitative knowledge.

With respect to its inference capabilities, the problem solving method must be able to process the decisions entered by the grower. It has been argued that if the problem solving method can process settings that take the form of desired momentaneous, average or cumulative bounds on the decision variables (applicable for specific periods within a day) this demand will be met.

Furthermore, it was concluded that the grower must be able to analyze how his settings influence other variables which he considers indicative. This type of analysis requires that settings entered on variables representing crop characteristics must be propagated to variables representing climate characteristics and vice versa. Similarly, the influence of decisions with respect to momentaneous variables must be observable on cumulative variables (and vice versa).

Additionally, the representation method should be modular such that models can be easily added and/or replaced. This requirement is most easily realized if a declarative problem solving framework is used.

With respect to speed of execution, the inference method must be able to generate constraints with sufficient frequency. These constraints serve as input to the second phase of the autonomous control task. They are restrictions on the values that the quantitative variables, available in the models used by setting calculation mechanisms, can take.

6.6.2 The problem solving method for the inference engine

In the remainder of this work constraint reasoning (together with a constraint-based knowledge representation) will be used as the problem solving method for the inference engine. Although constraint reasoning *may* fulfill the above criteria, it cannot be determined beforehand that it will. Especially compliance to performance demands are difficult to estimate beforehand.

The choice of a problem solving method within this type of problems is not an easy one[139] (*e.g.* Camard *et al.*, 1994; Kokeny *et al.*, 1996). In the first place, there are many *additional*[140] attributes of the problem that may be of importance, determining their relative weight is difficult. Secondly, aspects like: experience with, and knowledge of appropriate techniques also plays an important role. Consequently, the choice of a problem solving method is subjective one, and even if the selected

---

[139] The CHIC-2 report (Kokeny *et al.*, 1996) on methodology issues in large scale combinatorial optimization problems provides helpful guidelines for practitioners in this area.

[140] Additional in the sense that they have not been mentioned in section 6.6.1. To name a few: the number of variables, their domains, the types of relationships between the variables (whether they are linear/non-linear, cyclic/non-cyclic, etc.), the way uncertainty and preferences are implemented, etc..

technique has proven to be sufficient at the end of the project, there might be better ones[141].

One activity in selecting a suitable problem solving method is trying to model a prototype problem in the terminology of the problem solving method. If this activity can be carried out successfully, the technique may be a good candidate, however, upscaling may still inhibit its practical application. With respect to our choice for constraint reasoning, chapter seven and eight of this thesis may be seen in this light.

Another aspect in the selection process is the question whether fitting of the method to the particularities of the problem is easily possible. Such adjustment may imply extensions of the method and the selection of a suitable representation and implementation structure. In chapter seven it will be shown that the application of constraint reasoning as the principal problem solving method in the inference engine required both adaptations: the Newton interval method was integrated into the core methods of constraint reasoning and these methods were implemented in the mold of an object-oriented framework.

---

[141] For instance, because of enhanced insight in the nature of the problem during the project, or because of increased computational power of computer hardware allowed the application of techniques that were considered to be inappropriate at the moment of choice.

7. GREENHOUSE OPERATIONAL PRODUCTION MANAGEMENT FORMULATED AS A CONSTRAINT REASONING PROBLEM

7.1 Introduction

Chapter six outlined a design of a supervisory control system to support the grower in his operational management task. The modules identified in this design vary in complexity and in how far they can be called scientifically innovative. Since it is unfeasible – within the scope of this work – to produce a complete working system[142] on the basis of the proposed design, it is necessary to choose how to proceed further. One possibility is to consider the design to be the endpoint of this research. Doing just so leaves many interesting questions unanswered and causes this research to be a predominantly mental exercise. Another possibility is to select what is expected to be a feasible, challenging and scientifically innovative part and investigate that issue further. The latter approach is pursued here.

It has been concluded in the previous chapter that the inference engine can be considered the most scientifically innovative part of the overall design. Therefore, the inference engine will be explored further in this chapter. It will be shown that:

- the inference task carried out by the engine can be formulated as a constraint reasoning problem, and that
- a prototype engine can be constructed to solve such problems. The workings of this prototype will be illustrated on some examples in chapter eight.

The chapter is structured as follows. First, in section 7.2, constraint reasoning will be described. The terminology explained there, will be used in section 7.3. In this section it will be illustrated how the task of inference engine can be formulated as a constraint reasoning problem. Next, in section 7.4, an implementation of a prototype inference engine will be discussed, the operation of this prototype will be explored later. Finally, in section 7.5 some of the particularities of the chosen framework will be discussed.

---

[142] Producing a working prototype shows that the proposed design is feasible, and that it *can* support a grower in his operational management task. Whether the ultimate system (after substantial addition development) will actually be embraced by growers requires even more work (*e.g.* training of growers, installation at a site, etc.).

7.2 Description of constraint reasoning

7.2.1 Informal presentation

7.2.1.1 Classification and background information

Constraint reasoning[143] is a formalism of representation and resolution of problems.

A constraint reasoning problem is represented in terms of variables and constraints on the variables. A constraint can intuitively be thought of as a restriction on a space of possibilities (Van Hentenryck and Saraswat, 1997). Constraints restrict the possible values that variables can take, thereby representing some kind of partial information about variables in the problem domain.

In constraint reasoning applications (like in other declarative problem solving methods) the definition of the problem is clearly separated from the methods used to solve the problem, this guarantees that the problem to be solved is precisely defined.

Constraint reasoning makes use of two types of deductive methods namely: *i*) the class of procedures that are related to consistency enforcing, and *ii*) the class of procedures that control and/or guide the process of searching for (a) solution(s). From the latter it can be concluded that constraint reasoning belongs to the class of generative problem solving techniques. Generative techniques[144] explore the solution space little by little. This means that at each step a partial solution is extended by tentative assignments (or more generally: adding constraints) until a solution is found or until it becomes clear that the exploration will not be successful and retrieval of another partial solution is needed. The way solutions are generated depends on the application, in some applications solutions will be generated automatically by the search algorithm, in others the user will be partially involved in the search process. In theory, the main advantage of a generative problem solving technique is that an optimal solution cannot be missed (Aarts and Lenstra, 1998). In practice, however, time and resource limitations require that the search process must be bounded, therefore not all alternatives are explored.

Constraint reasoning has originated from the field of Artificial Intelligence (AI) as a special problem solving methodology. The articles of Montanari (1974) and Waltz (1975) may be seen as its genesis. Today, constraint reasoning can be considered as

---

[143] The terms 'constraint (logic) programming' or 'constraint satisfaction' are also used. Throughout this chapter the term constraint reasoning is preferred over constraint satisfaction because the reasoning or inference mechanisms are discussed here. The 'satisfycing' of a constraint network depends on attributes specific to the problem instance at hand and, ideally, not on the specifics of the inference method used.

[144] As opposed to iterative techniques like genetic algorithms and simulated annealing which explore the search space starting from (a) complete (set of) solution(s) and (try to) improve the best solution found so far.

a well circumscribed field of research because it shows all the necessary characteristics[145]. The field of constraint reasoning may best be positioned between the fields of Operations Research and Artificial Intelligence, since it uses problem solving methods from both.

Constraint reasoning techniques have successfully been applied in combinatorial problems like planning and scheduling applications. For instance, Simonis and Cornelissen (1995) and Anonymous (1997) present practical applications in which constraint reasoning is the central problem solving method. These authors also claim that constraint reasoning was the best tool for their specific problem. As compared to techniques from operations research, constraint reasoning techniques generally perform best when the domains of the variables are discrete (or non-numerical), the constraints are not linear (in case of numerical constraints) and the problem specification contains many local or specific constraints (Chamard *et al.*, 1994; Kokeny *et al.*, 1996).

Recently, constraint reasoning is also successfully being applied in problems in which the domains of the variables are continuous. For instance, Van Hentenryck *et al.* (1997a) show an application of interval analysis techniques within a constraint reasoning setting on numerical benchmarks. They show that interval analysis techniques within a constraint reasoning setting successfully solves benchmarks that have not yet been solved by other means. Interval analysis techniques within a constraint reasoning setting will also be used here, they will be elaborated upon later in this chapter.

7.2.1.2 Basic components: variables and constraints

A constraint reasoning problem can be specified through the variables, their respective domains, and the constraint relations it includes.

Variables generally represent meaningful concepts in the problem domain[146] and are characterized by the type of values they can take on (*i.e.* their domain). The domain of a variable can be continuous (*e.g* `John's_Length ∈ [1.70,2.10]`) or discrete (*e.g* `John's_Length ∈ {fairly_short, average, fairly_long, long}`). Discrete domains can be either symbolic or numeric. Discrete numeric domains can be finite (*e.g.* the set of integers between 1 and 9) or not (*e.g.* the set of natural numbers (N)).

A constraint either expresses a relationship between two or more variables or simply a restriction on the values that a variable can take. The latter type are called unary

---

[145] It has a clearly identifiable research community. There also exists an international journal "Constraints" and there are a number of yearly held symposia and workshops that have constraint reasoning as their central theme.

[146] Note that in this thesis the term 'domain' is being used in two ways. In context of knowledge subjects, it approximately indicates a 'field' or 'specialty'. In the context of a variable, it refers to the set of candidate values for the variables.

constraints. Unary constraints typically represent either the decisions of the problem owner, or instantiations during the course of a search process. Regarding the former: binary, ternary,..., and n-ary constraints can be identified. Owing to the diversity in the domains of the variables, constraints can be expressed in many different ways. Some examples of constraint relations are:

- a unary constraint reducing the domain of a continuous variable from `[18.5, 21.5]` to `[18.5, 20.2]`, or the input of a measurement: $T_{\texttt{outside,t=10:00}} := 15.3$.
- binary constraints between continuous and symbolic variables (representing the same concept) expressed as lists of valid 2-tuples that map symbolic values onto their numeric (interval) counterparts (*e.g.* `{(average, [1.70, 1.80]);(fairly_long, [1.80, 1.90]);(long, [1.90, 2.00])}`).
- ternary constraints between continuous variables expressed as mathematical equations (*e.g.* $z = 2x^2 + 5y$);
- n-ary constraints between *n* variables expressed as lists of valid *n*-tuples (*e.g.* `{(red, yellow, blue, green); (red, yellow, green, green);` ...`; (yellow, green, blue, blue)}`);

7.2.1.3 Prototypical resolution methods

Constraint reasoning typically employs two types of inference methods, namely, consistency enforcing and search. These methods are normally in some way intertwined. That is, the consistency enforcing procedure is applied during the generation of a solution (which is carried out by the search mechanism; this mechanism may also include some kind of user interaction). The search methods can among others be aimed at finding whether a solution exists, finding one or finding all alternative solutions.

In its most general form consistency enforcing or constraint propagation consists in deducing new constraints from existing constraints. This includes removing inconsistent parts of the domains of the variables by checking these domains against the applicable constraints. Or, stated differently, the domain of a variable is 'reduced' or narrowed by the consistency enforcing procedure. For instance, consider the integer variables x_1 with domain [2,...,7] and x_2 with domain [3,...,9] and the constraint c_1: $x_1 > x_2$. Consistency enforcing results in the reduction of the domain of x_1 to the values [4,...,7] and reduction of the domain of x_2 to [3,...,6]. The search mechanism (with the objective to find all solutions, assuming there are no more constraints), results in the discovery of ten solutions (*i.e.* {(4,3); (5,3); (5,4); ...; (7,6)}). Consistency enforcing methods are applied locally, that is, each constraint (or each class of constraints) is propagated independently of the existence or non-existence of other constraints. This property enables the efficient combination of multiple constraint propagation techniques associated with different classes of constraints (Kokeny *et al.*, 1996).

There are numerous consistency enforcing and search methods; most are tailored towards dealing with problem classes that manifest particular properties. These methods typically aim at reducing the (inherent) computational complexity of search problems. One of the most common constraint reasoning procedures is 'Forward Checking' (*e.g.* Haralick and Elliot, 1980; Kumar, 1992; Tsang, 1993). Forward Checking (FC) is a prospective technique based on backtracking that loops through the set of variables and applies consistency enforcing after each variable instantiation (*i.e.* value assignment). For instance, in the above example the FC algorithm may first instantiate x_1 to 4, next the consistency enforcing mechanism will remove the values 4, 5, and 6 from the domain of x_2. It will then instantiate the next (and final) variable x_2 to the (only remaining) value 3. When the method has found a solution, or when it encounters an inconsistency, it backtracks and alternately tries to instantiate new values (*e.g.* next the value '5' can be instantiated for variable x_1).

The above concludes the informal presentation of constraint reasoning, the remainder of section 7.2 describes constraint reasoning in considerable more detail. The reader who is mostly interested in greenhouse crop production, may want to skip the remainder of this section and continue in section 7.3 where the application of constraint reasoning on the problem of greenhouse crop production will be discussed.

7.2.2 Definitions

7.2.2.1 Basic definitions

Problems expressed as constraint reasoning problems consist of variables, domains and constraints. A constraint reasoning problem *P* can be defined as follows (*e.g.* Mackworth, 1987; Tsang, 1993):

$P = (V, \mathrm{D}, \mathrm{C})$ where:

- C is a finite set of constraints $\{c_1, c_2, \ldots, c_n\}$.
- *V* is a finite set of variables $\{x_1, x_2, \ldots, x_n\}$.

  The variables that are constrained by constraint c_i are called the ***argument*** or subject variables of c_i. Constraints c_i and c_j can have overlapping sets of argument variables.
- D is the problem domain, $\mathrm{D} = \mathrm{D}_{x_1} \times \mathrm{D}_{x_2} \times \ldots \times \mathrm{D}_{x_n}$.

  The domain D_{x_i} of variable x_i is the set of all possible domain elements that can be assigned to x_i. A domain element or value can in principle be any object, however, we restrict ourselves to numerical, interval and symbolic values. A numerical value is an integer, a real or a rational number; an interval value denotes a closed continuous interval [a, b] with a as its lower and b as its upper bound; a and b are numerical values. A symbolic value denotes a symbol (*e.g.* 'Monday', 'high', 'red').

A problem *instance* P' is a specific instance of a constraint reasoning problem P with a fixed set of the variables V', with their initial domains D' and a fixed set of constraints C'. A constraint reasoning problem P is *dynamic*[147] when a problem instance P' changes into P'' through the addition or retraction of constraint(s) or variable(s).

A *label* is defined as a variable-value combination $(x_i, A(x_i))$, that represents an assignment to variable x_i of a value $A(x_i)$ from the domain D_{x_i}. A *compound label* is a simultaneous assignment of values $\{ A(x_{y_1}),..., A(x_{y_{|Y|}})\}$ to a set of variables Y, where: $Y \subseteq V'$ (Tsang, 1993). A compound label is also called an *instantiation* (Alliot and Schiex, 1994). A *complete* instantiation is an instantiation for which $Y = V'$.

A *constraint* c_i on a set of variables X_{c_i} is *extensionally* defined as a set of compound labels A_{c_i} for the argument variables X_{c_i} in the constraint[148]. A constraint is *intensionally* defined as any function f over X_{c_i}, that is: $c_i = f_{c_i}(X_{c_i}) \rightarrow Bool$. Unless explicitly stated an intensional constraint representation will be used in the sequel.

An instantiation A *satisfies* a constraint c_j $(A \vDash c_j)$, if the values $\forall x_i \in X_{c_j}$ in A, applied to the function f_{c_j} evaluates to T (true). An instantiation that does not satisfy a constraint, is said to *violate* the constraint.

An instantiation A of the variables $Y \subset V'$ is *consistent* if it satisfies all constraints c_i in C' for which the argument variables X_{c_i} are all in Y. A *solution* S of P' is a consistent instantiation A of the variables Y, where: $Y = V'$. An instantiation A of the variables Y, where: $Y = V'$ *partially satisfies* C' when it satisfies at least one constraint c_i in C'.

7.2.2.2 Definitions with respect to interval variables, domains and constraints

In our common sense notion of the word *value* we presuppose at least two properties: firstly, every value is considered unique and indivisible, and secondly, assuming a certain precision[149], we can determine whether two values are identical or not. Regarding interval values the the above notions need to be defined. In the following I

---

[147] Planning and scheduling are temporal reasoning problems that deal with the dynamics of decision making. However, an implementation does not need to reflect this property, for instance one may look at every problem instance individually *i.e.* in isolation of previous ones.

[148] *E.g.* an extensionally defined constraint c_a over the variables x_1, x_2, and x_3: ({(x_1, 'red'), (x_2, 'yellow'), (x_3, 'blue')}; {(x_1, 'green'), (x_2, 'yellow'), (x_3, 'blue')}; {(x_1, 'red'), (x_2, 'yellow'), (x_3, 'green)}), could represent the intensional constraint: f_{c_a}: $x_1 \neq x_2 \neq x_3$, if the initial domains are: x_1 = { 'red', 'green'}; x_2 = {'yellow'}; x_3 = {'blue', 'green'}.

[149] The number of significant digits of a real number.

conform to the definitions of Moore (1979), Hansen (1992) and Van Hentenryck *et al.* (1997a).

An interval $[l,u]$ represents a set of real numbers $\{r \in \mathrm{R} \mid l \leq r \leq u\}$ in which l and u are real values. The width of the interval $[l,u]$ is defined as: $w = |u - l|$. An *interval value* I is an interval $[I_l, I_u]$. In finite computer precision the smallest possible interval I containing the real number r is defined by $[I_l, I_u]$, where I_l is the largest value of type float[150] smaller than or equal to r and I_u is the smallest value of type float greater or equal to the real number r.

An *interval variable* X is a variable which domain D_X is an ordered set of disjoint interval values I_i. The set of interval values I_i is usually generated on the basis of a single initial interval I_{D_x} and a *discretisation value* (λ) associated with the variable. The discretisation value λ can also be used to define equality, that is, two interval values A and B are considered identical when $\mathrm{MAX}(|A_l - B_l|, |A_u - B_u|) \leq \lambda$. The size of the discretisation value will normally be chosen on practical grounds.

An important concept in interval arithmetic is the notion of *interval extension*. The interval function $F : \mathbf{I}^n \rightarrow \mathbf{I}$ is an interval extension of the function $f : \mathrm{R}^n \rightarrow \mathrm{R}$ if and only if $\forall\ I_1 \ldots I_n \in \mathbf{I} : r_1 \in I_1, \ldots, r_n \in I_n \Rightarrow f(r_1, \ldots, r_n) \in F(I_1, \ldots, I_n)$; in which I denotes the set of interval values. Similarly, the interval extension of a constraint C can be defined; a constraint: $C{:}\mathbf{I}^n \rightarrow$ *Bool* is an interval extension of a constraint $c{:}\mathbb{R}^n \rightarrow$ *Bool* if and only if $\forall\ I_1 \ldots I_n \in \mathbf{I} : r_1 \in I_1, \ldots, r_n \in I_n \Rightarrow [c(r_1, \ldots, r_n) \in C(I_1, \ldots, I_n)]$ (Van Hentenryck *et al.*, 1997a).

---

[150] Implementation specifics regarding the precision of this floating point number (with a certain `single`, `double` or `extended` precision) do not matter at this point, although it is important when developing a constraint language. In practice such a language has to remain computationally sound therefore outward rounding should be applied.

Using this concept, interval extensions of functions defined in $\mathbb{R}$ can be defined. The basic arithmetic operations +, -, × and ÷ are defined as follows (Moore, 1979):

$$[a,b] \oplus [c,d] = [a+c\ , b+d]$$

$$[a,b] \ominus [c,d] = [a-d\ , b-c]$$

$$[a,b] \otimes [c,d] = [\textsc{min}\ (a\times c\ , a\times d\ , b\times c\ , b\times d),\ \textsc{max}\ (a\times c\ , a\times d\ , b\times c\ , b\times d)]$$

$$[a,b] \oslash [c,d] = [a,b] \otimes (1/\,[c,d]),\ \text{where: } 1/\,[c,d] = [1/d\ ,\ 1/c],\ \text{if } c>0 \text{ or } d<0.$$

In the above rules the division by an interval containing zero was excluded. However a definition which includes division by an interval containing zero is often useful[151], Hansen (1992) describes $[a,b] \oslash [c,d]$, in case $c \le 0 \le d$, as follows:

$$\begin{array}{lll} [a,b] \oslash [c,d] = & [b/c, \infty] & \text{if } b \le 0 \text{ and } d = 0 \\ & [-\infty, b/d] \cup [b/c, \infty] & \text{if } b \le 0 \text{ and } c < 0 < d \\ & [-\infty, b/d] & \text{if } b \le 0 \text{ and } c = 0 \\ & [-\infty, \infty] & \text{if } a < 0 < b \\ & [-\infty, a/c] & \text{if } a \ge 0 \text{ and } d = 0 \\ & [-\infty, a/c] \cup [a/d, \infty] & \text{if } a \ge 0 \text{ and } c < 0 < d \\ & [a/d, \infty] & \text{if } a \ge 0 \text{ and } c = 0 \end{array}$$

For a complete set of rules for this so-called 'extended interval arithmetic', including the basic interval arithmetic functions on infinite and semi-infinite intervals, the reader may consult Hanson (1968).

Of course, the above functions could be defined differently, however these definitions provide the *optimal* interval extensions for the basic arithmetic operations (Moore, 1979). Optimal interval extensions are defined as the interval extensions that return the smallest possible interval containing all real results.

Finding efficient optimal interval extensions for classes of functions (*e.g.* polynomials) defined over $\mathbb{R}$ is a major research area in interval research. The reason why finding efficient optimal interval extensions is important, stems from what has been called 'the dependency problem' (*e.g.* Hansen, 1992). Consider the subtraction of interval $X = [a, b]$ from itself. Using the subtraction rule of above, results in $[a - b,\ b - a]$ and not $[0,\ 0]$ (unless $a = b$). In general, whenever a variable is present more than once in an interval computation, the dependency problem occurs because each occurrence is treated independently. This independent treatment causes widening of the computed intervals and makes it more difficult to obtain sharp re-

---

[151] Extended definitions are useful if the result of such a computation is to be intersected with already available domains. Consider *e.g.* the constraint C: $X_1 = X_2 \oslash X_3$ and $X_1 = [0, 5]$, $X_2 = [4, 8]$, $X_3 = [-4, 2]$. Application of the constraint first results in $[-\infty, -1] \cup [2, \infty]$, combining this result with the original domain of X_1 leads to $[2, 5]$. Without the extended definition no domain reduction could have been attained.

sults in the calculations. Techniques to reduce dependency problems can for instance be found in Hyvönen (1992) and Hansen (1992). One of the more common techniques is to rewrite functions in a form that suffers least from this problem (*i.e.* minimizing the number of occurrences of each variable), for instance in:

$$4X^3 + 11X^2 - 6X + 12 \equiv X(X(4X + 11) - 6) + 12$$

or

$$\frac{X - Y}{X + Y} \equiv 1 - \frac{2}{1 + \frac{X}{Y}} \cdot$$

If an interval variable occurs only once in an equation it cannot give rise to dependency (like in the rewritten form of the above example). To avoid the dependency problem, the following definitions (Hansen, 1992) are also provided ($X = [a, b]$):

$$X^n = \begin{cases} [1, 1] & \text{if } n = 0 \\ [a^n, b^n] & \text{if } \{a \geq 0\} \text{ or } \{a \leq 0 \leq b \text{ and } n \text{ is odd and positive}\} \\ [b^n, a^n] & \text{if } b \leq 0 \\ [0, \text{MAX}\,(a^n, b^n)] & \text{if } a \leq 0 \leq b \text{ and } n \text{ is even} \\ [-\infty, \infty] & \text{if } a \leq 0 \leq b \text{ and } n \text{ is odd and negative} \end{cases}$$

7.2.3 Inference within constraint reasoning

Much research in the field of Artificial Intelligence has been devoted to the area of problem solving. Problem solving on computers, in part, means looking for an efficient and effective inference method (algorithm) that can solve the problem already represented in a given format. Finding whether a constraint reasoning problem is consistent (*i.e.* has at least one solution) can be difficult[152]. In many cases one has to rely on algorithms that are based upon searching through all possible combinations of domain elements. In constraint programming literature several improvements of this search process have been reported (*e.g.* Gaschnig, 1977; Haralick and Elliott, 1980 in: Freuder and Wallace, 1992). These improvements can be classified as retrospective or prospective extensions to the basic backtracking algorithm. Kumar (1992) and Nadel (1989) discuss algorithms that are extensions to backtracking for constraint reasoning problems with finite domains. The general characteris-

---

[152] In its worst case the class of constraint reasoning problems is NP-complete, which means that finding an efficient and generally applicable algorithm for the problem class is likely to be impossible. Fortunately, research has resulted in methods to overcome this fundamental problem for many *specific* cases. Also, clever representation of the problem can be crucial for the tractability of a problem solving process (*e.g.* Tsang, 1993; Nadel, 1990).

tic of these algorithms is that in the average case complexity bounds[153] (and for some representations also the worst-case complexity bounds[154]) are being reduced significantly. The key idea behind these algorithms is either reducing redundant constraint checks or pruning values that do not meet local consistency criteria. A commonly used prospective technique is forward checking (*e.g.* Kumar, 1992; Prosser, 1993; Bacchus and Grove, 1995;) which is a combination of backtracking and local consistency enforcing techniques.

In this section forward checking will be explained in detail. Because of the locality principle (Steele, 1980 in Kokeny *et al.*, 1996) it is possible to include various local consistency enforcing methods that are specifically tuned to different constraint types (*e.g.* qualitative constraints, quantitative constraints, etc.). These variations will also be discussed.

The implementation of forward checking (together with the implementation of the definitions related to interval arithmetic) provides the means to show how constraint reasoning can be applied in the domain of crop production.

7.2.3.1 Forward checking

Figure 7-1 shows a generalized version[155] of the forward checking (FC) algorithm for finding one solution. The search process consists of sequentially instantiating the variables in an order specified by an ordering heuristic[156]. The boxed activities in Figure 7-1 represent the different procedures on which the forward checking is based. These procedures work either on problem variables or on constraint sets. The procedural steps within the FC algorithm will be explained in short. The consistency enforcing steps (*i.e.* step 2 and 6) will be explained separately in section 7.2.3.2.

---

[153] The complexity measure indicates the amount of work with respect to the size of a problem class (Garey and Johnson, 1979).

[154] In tree-shaped constraint problems arc consistency means global consistency (Freuder, 1982). However, constraint networks are usually not tree-shaped (*i.e.* cycles can be detected) which means that additional search, intertwined with consistency checking and domain filtering (as in the forward checking algorithm) is needed to find a complete solution (or to conclude that no solution exists).

[155] Note that there are many variants of forward checking, Kumar (1992) called the above version "really full lookahead", because it propagates every domain change until quiescence. See also Wallace (1993) for a discussion on the performance of some variants.

[156] If no variable ordering heuristic is applied the ordering takes place in a non-deterministic manner and depends on the progress of the reasoning process itself.

Applicable constraints (step 1 and 5)

Search algorithms (like FC) work on sets of constraints: the constraint stack. In step 1, the applicable constraints are all the constraints C^* in the problem instance P^* that constrain the problem variables $V_{p'}$. For step 5 the applicable constraints C_{v_i} are the constraints that influence the selected variable v_i.

Variable selection (step 3)

A good variable ordering can significantly improve the speed of reasoning because pruning of the search space occurs earlier in the search process. Variable ordering can be carried out in both a static as well as dynamic fashion. In a static ordering the order of the variables is computed before the search starts, while in a dynamic ordering the order of the variables is modified during the search process. Common heuristics for variable ordering are:

- choose the variable with the smallest domain first,
- choose the most constrained variable first (*i.e.* the one belonging to the greatest number of constraints), and
- use a fixed ordering (in the static case only. The ordering is based on a logical criterion that usually requires a deeper understanding of the constraint problem and that cannot be deduced from the constraint relations automatically).

In the prototype described later, the third option has been chosen.

Domain preparation and value selection (step 4 and 8)

For interval domains, the selection of domain elements is preceded by a discretisation step. During this step the domain of the variable is discretisized using discretisation value (λ_v) for the variable v. The discretisation value λ_v depends on the granularity needed in the application. The domain D_v is discretisized into n interval values $I_1 \ldots I_n$ by taking the lower bound (D_ℓ) of D and adding the discretisation value of the variable to it. The first discretisized value becomes: $I_1 = [D_\ell, (D_\ell + \lambda_v)]$, the subsequent n interval values become: $I_n = [(D_\ell + (n-1) \times \lambda_v), \textsc{min}\{(D_\ell + n \times \lambda_v), D_u\}]$ until the upper bound (D_u) of D is superseded: $(D_\ell + n \times \lambda_v) > D_u$.

Recovering intermediate results (step 7)

Intermediate domain changes are logged during the domain filtering procedure. Whenever an inconsistent domain is detected the FC algorithm has to undo all domain changes since last value instantiation using these logged changes.

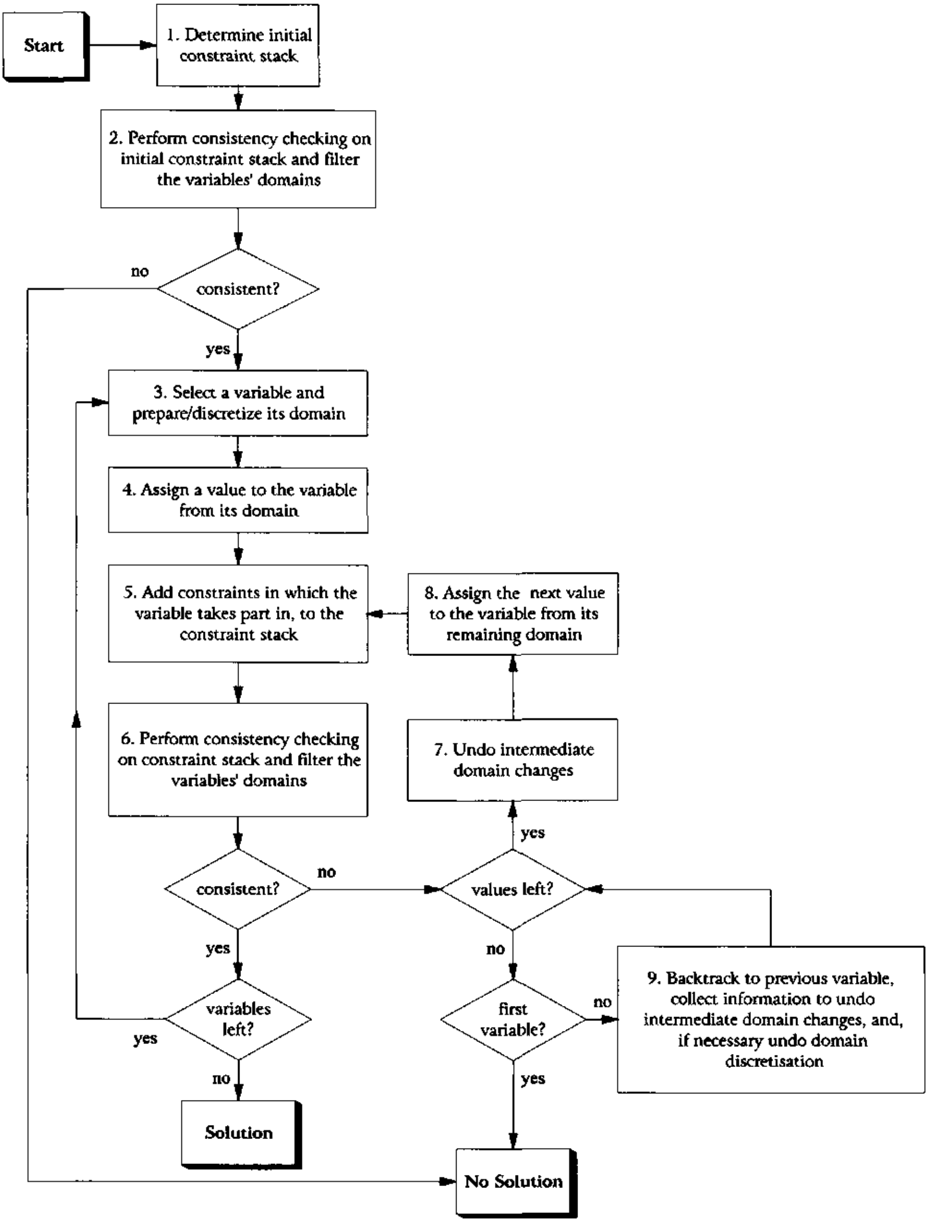


Figure 7-1 Forward checking procedure.

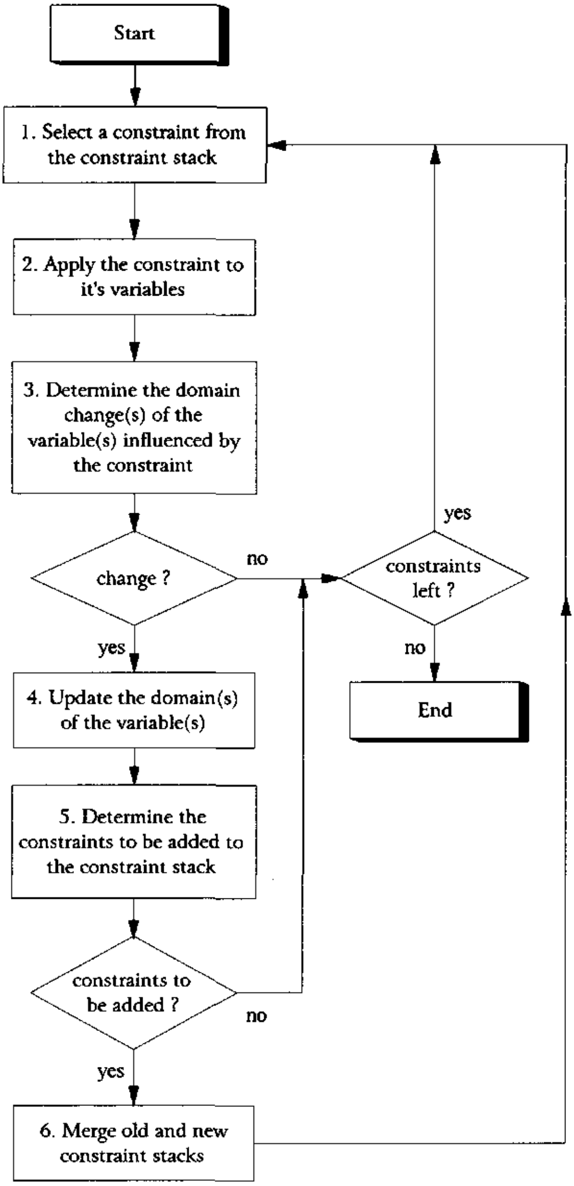


Figure 7-2 Arc consistency enforcing algorithm.

Backtracking (step 9)

Whenever a local inconsistency is detected and all values of the last instantiated variable are inconsistent, backtracking is needed to a previous instantiation. If further backtracking, that is, to previously instantiated variables, proves to be necessary variable discretisation has to be undone as well. For the variable of such previous instantiation a new value must be chosen if there is any left.

7.2.3.2 Local consistency enforcing with in forward checking

Consistency enforcing is an important part of the constraint reasoning paradigm. Consistency enforcing or constraint propagation removes inconsistent values from the domains of the variables in the problems specification. There are various kinds of consistency enforcing methods that differ in the amount of consistency they enforce (*e.g.* Kumar, 1992; Tsang, 1993). Here, only local or 'arc' consistency is considered, since it is most commonly used in real world applications. Local consistency methods ensure that the domains of the variables are consistent w.r.t. the constraints they participate in. These methods have a local view in that they look at each constraint individually. In general this means that if a constraint network is locally consistent, it does not imply or prove that a (global) solution exists.

Figure 7-2 shows the general procedure for local consistency checking and domain filtering. This procedure is an extension of the arc consistency enforcing algorithm (AC-3) that has been proposed for binary constraints (*e.g.* Tsang, 1993; Kumar, 1992). As can be seen in the figure this consistency enforcing algorithm works on lists of constraints, the boxed activities are sub-procedures and most of them work on individual constraints.

In an implementation, for instance through the use of object oriented programming techniques, it is possible to specialize the behavior of the different sub-procedures based on the characteristics of the constraint type. The steps 2 to 5 in Figure 7-2 are usually implemented as one procedure called '`revise`'. The behavior of this procedure is different for each constraint type, that is, the implementation contains special versions for unary, qualitative, interval and mapping constraints.

Select a constraint from the constraint stack (step 1)

Selecting a constraint from the constraint stack usually means getting the first of the stack. However, when constraint checks are computationally expensive determining an (a priori) ordering may be profitable. Unfortunately, the process of creating an ordering may sometimes be more expensive than the reduction in constraint checks the ordering brings about. In the implementation no specific constraint ordering has been applied.

`Revise` method for unary constraints (step 2 to 5)

Unary constraints are constraints that include one variable. They represent a relation between a variable and its domain. The constraint describes which part of the domain is valid (*i.e.* which values are valid). Application of the constraint requires intersecting the domain of the variable with the valid values specified by the constraint. If the constraint reduces the domain of the variable all other constraints in which the variable participates may need to be added to the constraint stack.

Unary constraints are special in the sense that they need to be evaluated only once.

`Revise` method for symbolic or qualitative constraints (step 2 to 5)

There are many procedures that can be used to enforce consistency on variables in constraints that include variables with finite (countable) domains. A simple – albeit inefficient – way to check whether the values in the domains of the variables are valid is:

1. generate all possible compound labels on the basis of the present domains,
2. check every compound label against the constraint and store the labels that are valid, and
3. gather for each variable the set of values that are present in at least one valid label.

The latter sets represent the new domains of the variables. For each variable one should check if its new domain differs from its original. For each variable which domain is indeed reduced, the constraints in which the variable takes part should be collected (except from the one just tested, and all unary constraints) and must be placed on the constraint stack (if they are not already on it).

The above procedure only works well for constraints with few variables having (very) small domains. There are many ways to improve the above procedure. For instance incremental generation of the compound labels is a possibility since one only needs to find one valid label to determine whether a value belonging to the domain of a variable is consistent. Other kinds of improvements are also possible *e.g.* for numerical (integer) constraints a method based on branch and bound has been reported (Freuder and Wallace, 1992).

`Revise` method for interval constraints (step 2 to 5)

To enforce consistency on nonlinear equations, which are commonly found in (numerical) crop models, more sophisticated methods are needed. Here, an algorithm from the field of Interval Arithmetic is used within the `revise` method.

The idea to use interval arithmetic in a constraint reasoning setting goes back to Cleary (1987) and Davis (1987). Later, amongst others, Hyvönen (1992) and Older and Vellino (1993) investigated methods to represent constraints more efficiently (*e.g.* through Taylor expansions), and investigated methods to propagate interval constraints. Recently, Van Hentenryck *et al.* (1997a, 1997b) report a method that has been based on a combination of an interval extension of the Newton root-finding method and a domain splitting method.

Within the `revise` procedure an adaptation of the interval Newton algorithm presented in Hansen (1992, p.74-75) is used (the precise procedure can be found in Appendix 2). The algorithm is adapted to better deal with perturbed equations, which is necessary since every n-ary constraint includes multiple variables that have domains which are initially wide. The algorithm reduces the domains of the variables in the constraint to their minimal interval with respect to the fixed perturbance in the constraint equation. The algorithm has a number of properties[157] that makes it a good candidate to be used in constraint reasoning procedures.

---

[157] Proofs of these properties can be found in Hansen and Greenberg (1983), Hansen (1992) and Dinkel *et al.*(1988).

1. Every zero of the function f(x_n) w.r.t. x_i for in the initial interval X_0 of x_i will always be found and correctly bounded. In constraint terminology: the algorithm returns the smallest feasible interval domains for the argument variables $x_1..x_n$ in constraint c (w.r.t. the fixed perturbance in c). Constraints in the form of eqn.(1) should be rewritten in the form of eqn.(2). A gradient function or the partial derivative of f must be available for every variable in the constraint.
2. If there is no zero for f(x_n) in X_0 then the algorithm will prove this fact in a finite number of iterations. In constraint terminology: inconsistent domains will always be removed.
3. Rapid convergence to the smallest feasible domain for every variable in the constraint.

Since this algorithm has been embedded in a *local* consistency checking mechanism, one should realize that in general *global* consistency can only be realized after additional search.

$$c : x_n = g(x_1 ... x_{n-1}) \qquad (1)$$

$$f(x_1 ... x_n) = g(x_1 ... x_{n-1}) - x_n = 0 \qquad (2)$$

As in the `revise` methods described above, the constraints involving variables having domains modified in the previous step need to be added to the constraint stack (unless they are already on it).

`Revise` method for mapping constraints (step 2 to 5)

The final `revise` method couples variable which domains are qualitative with their quantitative (interval) counterparts. These binary constrains form the link between qualitative and quantitative knowledge. Mapping constraints always work on pairs of variables that represent the same concept, only their representation differs.

Table 7-1 Symbolic - numerical mapping for relative humidity using a qualitative scale of seven values.

symbolic value	***interval***
very low	[0, 60]
low	[60, 65]
moderately low	[65, 70]
average	[70, 75]
moderately high	[75, 80]
high	[80, 85]
very high	[85, 100]

Table 7-1 shows the relationship between symbolic and numerical values for humidity. The number of qualitative values, and boundaries of the intervals given in this table are stated only for the sake of the example. It is up to the experts (*e.g.* extension specialists, growers) to decide – on the basis of their expertise – what their real values should be.

The `revise` method checks for every value in the domain of the qualitative variable whether its corresponding interval overlaps with the domain of the interval variable. If it doesn't, the value is removed. Afterwards, the union of the interval counterparts of the valid symbolic values intersected with the domain of the interval variable, results in the new domain interval variable. If either domain has been changed, constraints in which the variables participate in, may need to be added to the constraint stack (like in the previous `revise` methods).

Update the constraint list (step 6)
The `revise` method results in a list of constraints that may need to be added to the constraint stack. Some of these constraints may already be on the list so this must be checked. Only constraints that are not yet on the list should be added (typically at the end of the list). This step may also be used to apply a certain constraint ordering. In such case the constraints are *inserted* in the list according to the ordering.

7.3 The inference engine modelled as a constraint reasoning problem

7.3.1 Introduction

Modelling the problem is an important aspect and the first step in solving the problem computationally. Modelling a problem can be defined as finding a representation of the critical properties of a problem within the context of a particular representation formalism. Finding a 'good' problem representation is not a trivial task (see for instance the discussions in: Nadel, 1990; Hartog and Beulens, 1993; Chamard *et al.*, 1994; Paltrinieri, 1994; Simonis and Cornelissen, 1995 and Kokeny *et al.*, 1996). As an example, Nadel (1990) discussed the N-queens[158] problem and showed various possibilities to model this (toy) problem in a constraint reasoning formalism. The main lesson here is that the choice of the problem representation can severely influence the performance of the implementation. Thus the representation in part determines the feasibility to solve a similar but larger version of the problem (*i.e.* more variables, values and constraints).

Forming the inference engine in a constraint solver requires dealing with the following key properties:

1. *goal of the inference engine*
   The goal of the inference engine is to generate the most appropriate output constraints for the setting calculation mechanisms in the light of: *i*) the settings (and preferences) of the grower, *ii*) the present and expected (future) state of the greenhouse–crop system, *iii*) additional (external) inputs, and

---

[158] Place N queens on a N×N chessboard in such a way that no queen threatens another queen.

iv) the available domain knowledge. The goal of the inference engine determines the way the inference methods within constraint reasoning are applied.

2. *output of the inference engine*
   The output of the system must be such that subsequent parts of the overall system can use it. This requires insight in the capabilities of the setting calculation mechanisms such that a clear formulation of how the output constraints for the setting calculation mechanisms should look and with which frequency they must be generated.
3. *various types and sources of input*
   The inference engine gets its input from various sources: the grower, the measurement system and various external agencies. Furthermore, this input has different properties, namely: facts, settings and predictions. These must all be stated in constraint reasoning terminology.
4. *state of the greenhouse–crop system over time*
   The inference engine needs a representation of the state of the greenhouse–crop system. For proper crop management it is believed that this representation should be dynamic, therefore part of the past and future state should also be present.
5. *representation of domain knowledge*
   Instrumental in the constraint reasoning process is the domain knowledge. This knowledge must be represented as constraints. The inference engine uses this knowledge to infer its output from the represented state, and the various sources of input.

These properties will be discussed in more detail below. However, before doing so the issue of how to deal with the dynamics of the problem will be discussed first. Finding a suitable representation for the variable 'time' is important, because it concerns each of the five aspects.

The remainder of this section is as follows: first the representation of time will be discussed, afterwards the five properties mentioned above will be discussed individually, and finally, the implementation of the inference engine will be explained.

7.3.2 How to treat time?

In finding a suitable representation for the variable 'time' requires that demands from two sources must be satisfied:

- intrinsic demands of the problem.
  Solving the problem successfully requires that inputs and outputs are dealt with appropriately. This means (among others) that the grower can enter input constraints on the decision variables that apply to specific time intervals or periods. Furthermore, (parts of) the state of the greenhouse–crop system must be represented in the system such that the grower can observe the pres-

ent, past and future state of the greenhouse–crop system. Finally, part of the domain knowledge consists of dynamic models relating a certain state at time *t* to its state at a previous time instance. This kind of relationships must somehow be represented as constraints.

- demands related to the inference method.
  This point refers to the way time must be represented in order that constraint reasoning techniques can appropriately deal with the resulting constraint reasoning problem. There may be several possible representations that lead to such constraint reasoning problems. Consequently, the implementor has a choice and should select w.r.t. criteria like: efficiency, extendibility, etc..

The approaches (chapter four) that address the problem of operational crop production management (or more specifically climate control) generally deal with the variable 'time' in a continuous manner. Here, the problem the inference engine has to deal with, is modelled as a *discrete* problem. Time is treated in a discrete and hierarchical manner (Figure 7-3). The hierarchy corresponds to the periods[159] identified in chapter six. Every period object contains attributes that represent the state of the greenhouse–crop system during the time interval in question. If the period object lays in the past, present or future, the attributes of this object represent the historical, actual or expected state of the greenhouse–crop system.

The lowest level in the time hierarchy is set to be one hour. This interval is in line with both the optimization horizon of the setting calculation mechanism for climate control[160], as well as the dynamics, or rates of change of the processes described in the domain models that are used by the inference engine[161]. The relations in the domain models are expressed as constraints and these constraints connect the components (or attributes) in the hierarchy (this issue will be elaborated upon in section 7.3.7).

Concluding: the discrete and hierarchical treatment of time presents quite a different view on the implementation of the system's problem solving task compared to most of the approaches considered in chapter four. However, once variables, domains and domain relations, etc. are placed in a discrete setting, the constraint reasoning methods of section 7.2 can readily be applied.

---

[159] Since the day-length varies slowly over the season, the duration of the day and night period will also vary.

[160] In this discussion only the setting calculation mechanism for climate control will be considered, since it is clearly more complex than the one for irrigation and nutrient control.

[161] In chapter 3 it has been discussed that the processes that are relevant for the inference engine may indeed be modelled with a timestep equal to or larger than one hour.

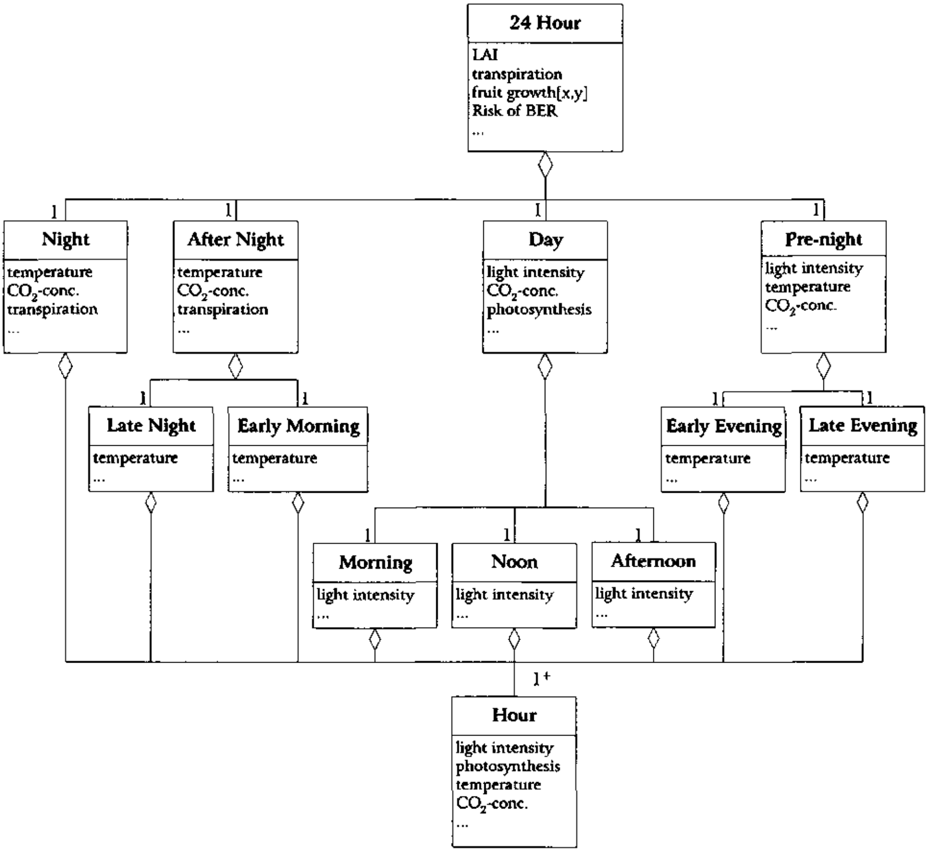


Figure 7-3 Diagram of the time hierarchy describing the 'part-of' relation between period classes[162].

7.3.3 The goal of the inference engine

The goal of the inference engine is to deliver output constraints or boundary values for the setting calculation mechanisms (Figure 7-4). Since the optimization horizon of the setting calculation mechanism for climate control is one hour, the output constraints could for instance be generated once an hour. At the cost of an increased computational burden, they can also be calculated with a higher frequency.

---

[162] Legend: the rectangles represent the classes that are part of the hierarchy; the diamond shaped connectors denote the `part-of` relation between classes; the numbers near the connected (bottom) classes depict how many of them are contained in the connecting (top) class. In each rectangle only a (very) few attributes per period class are shown (each attribute is a variable in the constraint reasoning framework).

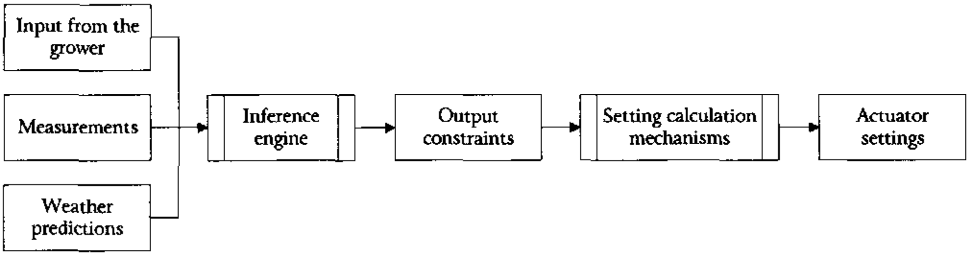


Figure 7-4 The autonomous control phase.

The goal of the inference engine is realized through search. The search process consists of finding a solution at the highest attainable preference level. Using the implementation of the Forward Checking algorithm discussed above, this requires solving multiple constraint reasoning problems. First, the inference engine generates a set of constraint reasoning problems on the basis of the preference distributions entered by the grower (*i.e.* one constraint reasoning problem for every preference level[163]). Next it finds the highest attainable preference level by applying the FC algorithm iteratively to a number of constraint reasoning problems. The set of constraint reasoning problems can *e.g.* be ordered in a dichotomic fashion, this ensures that if there are ten preference levels, the FC algorithm has to search for a solution in three or four problems.

There are also other possibilities to manipulate preference distributions (*e.g.* Martin-Clouaire 1993). The method suggested by Martin-Clouaire (1993) uses fuzzy set theory and deals with the preference distributions in their entirety. However, for this discussion, the method suggested here suffices.

7.3.4 The output of the inference engine

The output of the inference engine consists of the set of acceptable values for every domain variable in the inference engine. The setting calculation mechanism for climate control uses only a subset of these values. Typically, it uses the values for the variables that are available in both the setting calculation mechanism as well as the inference engine.

---

[163] As the goal of the inference engine is to deliver output constraints for the setting calculation mechanisms , it must do so under every circumstance. There are several ways to realize robust behavior. One may be to set up the lowest preference level in such a way that inference engine can always find a solution. Other options may be to change the search method and pursue partial satisfaction only (*e.g.* Freuder and Wallace, 1992).

The setting calculation mechanism for climate control will contain both the variables associated with the control devices (Table 6-1), as well as the variables included in the numerical models available in this subsystem. Models that should be included pertain to relatively fast processes, like: models describing the greenhouse climate, energy consumption, photosynthesis, and transpiration[164]. Therefore the setting calculation mechanism allows for an interface that consist of: *i*) the variables associated with the control devices, and *ii*) the state and rate variables that are part of the above models (depending on which of these variables are present in the inference engine).

The setting calculation mechanism should be able to deal with momentaneous, average and cumulative output constraints on the state and rate variables that make up the interface between both subsystems. For instance, the output constraints of the inference engine can be formulated as:

- momentaneous constraints that are applicable at each control instance of the setting calculation mechanism (*e.g.* $T_{air} \in [19.1, 19.8]$ degrees C), and
- average and cumulative constraints to be realized during periods within the optimization horizon of the setting calculation mechanism (*e.g.* $T_{air,\,hour} \in [19.5, 19.8]$ degrees C, or $\Sigma_{hour}\text{Transpiration} \in [50, 60]$ $g{\cdot}m^{-2}{\cdot}h^{-1}$).

Given the discrete treatment of time in the inference engine, these output constraints pertain to a specific time interval. For instance the momentaneous constraint $T_{air} \in [19.1, 19.8]$ degrees C, refers to the time interval 9:31–10:30, while for the interval 10:31–11:30, the inference engine may have generated the constraint $T_{air} \in [19.3, 20.5]$ degrees C.

7.3.5 Various types and sources of input

For the calculation of output constraints the inference engine gets input from three sources: the grower who has entered his settings, the measurement system that measures the actual state of the greenhouse–crop system and the weather agency that estimates the future external conditions.

7.3.5.1 Settings of the grower

The settings of the grower constitute the principal input for the inference engine. The settings of a grower are implemented as unary constraints on decision variables[165]. Decision variables can be quantitative or qualitative (symbolic) and apply

---

[164] In Tap (in preparation), the responsibilities of the setting calculation mechanism also include judgmental decision making, this is why models of slow processes (*i.e.* crop growth and development) are also included. It has been argued in previous chapters that judgmental decision making should be left to the grower.

[165] The *interface* variables are defined as the variables that represent meaningful concepts for the grower within the context of the supervisory control system. The *deci-*

Continued →

to a specific period. Input constraints on quantitative and qualitative variables can be momentaneous, average and cumulative w.r.t. the period for which they are set[166].

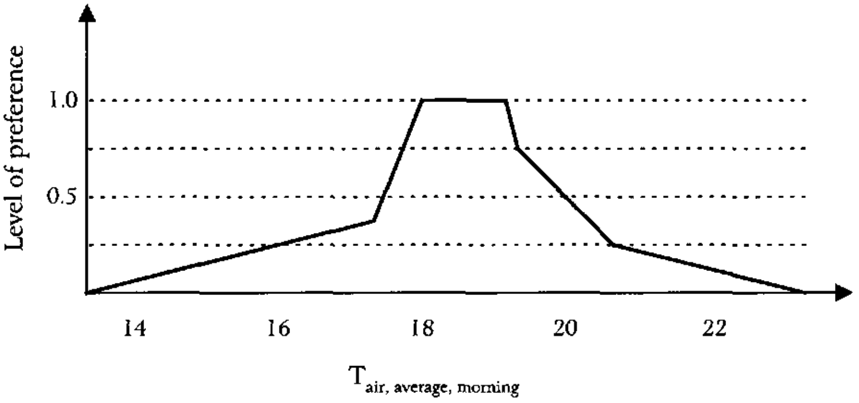


Figure 7-5 Example of an input constraint with preference distribution.

The input constraints of the grower involve preference distributions over the range of values a variable can take on. As already mentioned, the preference distribution supports the notion that decisions are not imperative and usually involve finding a compromise. The preference ordering is a means to automatically strive for the highest achievable preference level given external conditions. The preference ordering on the input constraints allows the grower to have his system behave differently at specific preference levels as can be seen in Figure 7-5, at low preference levels this input constraint will not matter much.

Finally, the decisions of the grower may also be formulated conditionally (*e.g.* `if` $T_{outside,t=t} > 15°C$ `then` $T_{air,\ t=t} > 18°C$), in which case the condition may evaluate positively, negatively or indeterminately (*e.g.* when $T_{outside,t=t} \in [14.5, 15.5]$ °C). The latter situation is caused by uncertainties in the weather prediction, hence, the

---

sion variables are the subset of the interface variables which the grower can constrain. The *control* variables are the subset of the decision variables that directly relate to the available control devices and which values are transmitted to the control systems at regular intervals

[166] From a common sense point of view one may note that all possible combinations are not equally good. For instance, requiring a particular average 24-hour photosynthesis rate may better be replaced by some kind of desired cumulative value for photosynthesis over the light period. However, from a computation point of view the above differences do not matter much.

resolution mechanism should supply appropriate default responses for the indeterminate case (*e.g.* not applying the constraint in this particular case)[167].

7.3.5.2 Measurements

The measurements constitute the second source of input for the inference engine. Measurements appear at the lowest (*i.e.* hour) level in the hierarchy (*e.g.* $T_{outside,\ t=11hour}$ = 15.3°C). They are preprocessed (*i.e.* accumulated, averaged, etc.) by the measurement module[168]. Like the settings of the grower, measurements can be seen as unary input constraints.

7.3.5.3 Weather predictions

Weather predictions are gathered from weather service bureaus a few (typically two or three) times a day. The module that processes the weather predictions estimates a sequence of quantitative data for the external variables based on the (qualitatively stated) predictions. These data necessarily contain some uncertainty. In the examples presented later, this uncertainty will be represented through intervals. Like the previous two kinds of input, the weather predictions can also be seen as unary input constraints.

7.3.6 Representation of the state of the greenhouse–crop system

The state of the greenhouse–crop system is represented in the attributes belonging to the period or time interval objects in the time hierarchy. A period object refers to a unique time interval and is constructed during the course of the season. Upon object construction an initial domain is assigned to every attribute (*i.e.* variable within the constraint reasoning framework) that belongs to the period object. Initially, this domain contains all possible values and will be reduced when constraints are added (or propagated during constraint reasoning). Domain reduction of the attributes belonging to a particular time interval object will naturally stop when the period object 'elapses'.

An attribute of a period class is a quantitative or qualitative variable that describes part of the state of the greenhouse–crop system during a particular period. The place of each attribute in the time hierarchy depends upon rate of change of the concept the attribute describes. For instance, the leaf area index is a slowly chang-

---

[167] This type of input requires a special consistency enforcing procedure that has not been discussed in the section 7.2.3.2. Since this type of input can be seen as a kind of 'syntactic sugar' on the basic mechanisms it will not be discussed further.

[168] Within the inference engine they can be further accumulated or averaged (*i.e.* several levels in the time hierarchy show multiple variables that represent the same concept).

ing aspect of the crop and is therefore placed at the 24 hour level. On the other hand, a concept like crop transpiration varies rapidly over the course of a day and should therefore be placed at a low (*i.e.* hour) level in the hierarchy. For concepts that are modelled – as opposed to concepts that are measured – the place in the hierarchy depends upon the timestep of the model that is used. For transpiration one could also use a model that calculates the transpiration rate with a timestep of a day (*i.e.* 24 hour), in this case the variable should be placed at the 24 hour level.

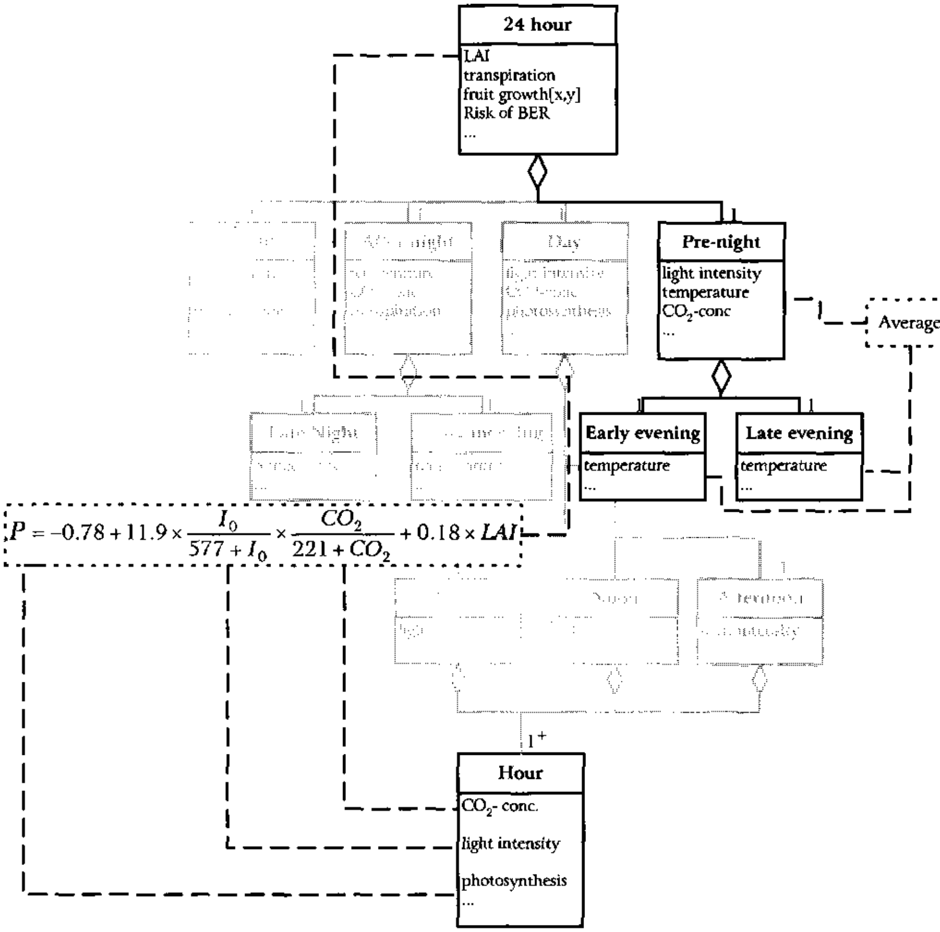


Figure 7-6 Two types of knowledge relations.

7.3.7 Representation of domain knowledge

Variables in the time hierarchy are connected through the relationships that exist between them. Two classes of relationships may be discerned (Figure 7-6). The first class consists of accumulation and averaging relationships that connect variables at more aggregated levels in the time hierarchy with their more detailed counterparts.

The second class of relationships are the domain models. These relations mainly connect variables at the same level although some variables come from more aggregated levels. This is can be seen in Figure 7-6 in which the photosynthesis (P) for a particular hour can be calculated using the equation shown. This equation relates the hourly Photosynthesis to the hourly CO_2-concentration, the hourly light intensity and the leaf area index (LAI) at the 24 hour level. The two classes of relationships can be represented as constraints.

A time-based datastructure provides a natural representation of the dominant data flows in the problem domain. That is, the measurements map to the lowest level period (*i.e.* hour), likewise, the settings of the grower apply explicitly to a specific period (*e.g.* temperature during *night* or *afternoon*, etc.).

7.4 The implementation of a prototype inference engine

The procedures for constraint reasoning (section 7.2.3), the arithmetic w.r.t. interval computation (section 7.2.2.2), as well as the functionality needed to: *i*) set up the time hierarchy, *ii*) generate the variables as part of the period classes and *iii*) define the constraints over these variables (section 7.3) have been implemented in Common Lisp.

Common Lisp is an object-oriented programming language (*e.g.* Keene, 1989; Steele, 1990) and is very well suited for prototyping since it contains features like automatic memory management and incremental compilation, it allows for dynamic definition of new constructs, and it has many (> 700) predefined functions (Norvig, 1992). Furthermore, it is available on every mainstream operating system–hardware platform combination.

7.4.1 Object oriented programming and design

7.4.1.1 Object oriented programming

The prototype has been implemented using object oriented programming and design techniques. The advantages of object oriented design and programming are manifold. The following may be the most prominent (Graham, 1994):

- Easier modelling of the real world. Modelling a problem in terms of *components* that have both structure and behavior seems to be very natural. The meaning and semantics of real world objects can be captured more effectively.
- Productivity improvement. Reuse of earlier software engineering efforts is considered a predominant advantage of object-oriented programming and design. However, this advantage has proven difficult to realize (*e.g.* Meyer, 1995). User interface builders are a classical example of code reuse. Reuse

has been mainly centered around code reuse, however, recently it was realized that design reuse in the form of 'design patterns' can also be of practical value (*e.g.* Gamma *et al.*, 1995; Schmidt, 1995).

In addition to the coupling between structure and behavior, object-oriented technology is generally centered around the following three attributes: inheritance, polymorphism and encapsulation.

Polymorphism

Polymorphism applies to the behavior of objects, *i.e.* to the way objects respond to a *message* that is sent to them. Messages[169] are calls to procedures that belong to the interface of an object. Messages to objects of different classes may share the same name although their implementation can be different, consequently, upon execution, the resulting "chain of events" will also be different.

Inheritance

Inheritance refers to the process of deriving attributes and behavior through class–subclass relations. For instance, given a class `Plant`, a *sub*-class `CAM-Plant` could be defined. The latter[170] inherits all the attributes and most of the behavior of the regular `Plant` class however, some behavior will be different. The class may also have additional attributes. In this particular case, the photosynthesis process (among others) will be different, therefore, the procedure or method that describes this process will have to be different. The standard method for `CalcPhotosynthesis` defined for the class `Plant` will be *overridden* with a new one (with the same name) belonging to the class `CAM-Plant`. All objects that are instances of the `CAM-Plant` class will apply this new method and behave differently with respect to this procedure compared to objects that are instances of the `Plant` class. Note that for behavior that is not overridden instances of both classes will behave exactly the same. Object-oriented programming languages have (internal) logics that determine the most applicable method for a given object.

Inheritance relations result in class hierarchies; subclasses relate to their parent classes through an *IS-A* relation. Other hierarchies are also possible, for instance, the *PART-OF* hierarchy (Figure 7-3), in such hierarchy objects (instead of classes) are related to one another.

Encapsulation

Encapsulation refers to the 'interface' of objects. The interface of objects is defined at the class level and it contains all messages to which objects of that class can re-

---

[169] In Common Lisp terminology messages are an approximation of 'generic functions'. The procedures that implement these messages for the different classes are called 'methods'.

[170] A class that represents certain types of cacti and succulents which behave in some aspects (stomata behavior) differently than 'regular' plants.

spond. Through encapsulation it is possible to hide implementation details from the object's use. Proper use of encapsulation facilitates maintenance and extension of a class library.

7.4.1.2 Design

With respect to design, programmers are encouraged to use established design patterns (Gamma *et al.* 1994). Design patterns can be seen as a kind of formalization of knowledge on how to achieve some design goal within an object-oriented way of working. Design patterns explicitly capture knowledge that experienced software developers already understand and use[171]. Because of the generic nature of the solution pattern applied, source code implementing design patterns can be understood more easily and have a better chance to be reused in later software development projects.

In the implementation the following design patterns are applied[172]: the Template Method design pattern, the Singleton design pattern and the Composite design pattern.

- The Template Method design pattern is useful in algorithms where some steps in the operation must be deferred to subclasses. Here it is used in the consistency enforcing algorithm or implement the `revise` procedure.
- The Singleton design pattern is useful in cases where a certain class can have only one instance which is a global point of access. Here the top of the period class hierarchy must be implemented in such way.
- The Composite design pattern is useful to create tree structures to represent part-whole hierarchies. Here the period class hierarchy is implemented using this design pattern.

7.4.2 The component structure of the implementation

The time-based datastructure (Figure 7-3) is constructed from a parent class or superclass **Period** and its subclasses. The class **Period** implements default attributes and behavior for all periods subclasses. Figure 7-7 shows some of the details of **Period** class in its class hierarchy using the OMT notation[173] (Rumbaugh *et al.*, 1991).

---

[171] *E.g.* Stefik and Bobrow (1986) and Rumbaugh *et al.* (1991) describe procedures to tackle software design problems. The results of these procedures can be seen as design patterns.

[172] The recognition of the implementation of what others (*e.g.* Gamma *et al.*, 1995) call design patterns in my design has only recently been perceived (*i.e.* after the prototype has been built).

[173] In the figures the following OMT notation has been used. Abstract classes are shown in italics. Abstract classes are used to define and implement common structure and behavior that is shared among the derived classes, they are (by definition)

Continued →

The **Period** class contains basic functionality needed in the time hierarchy, *e.g.* it contains facilities for adding new objects to an already existing hierarchy. During instantiation of new objects, begin and end times, predecessor and successor slots, etc. are automatically set. Predecessor and successor slots refer to elements at the same level, *e.g.* day, pre-night, night and after-night.

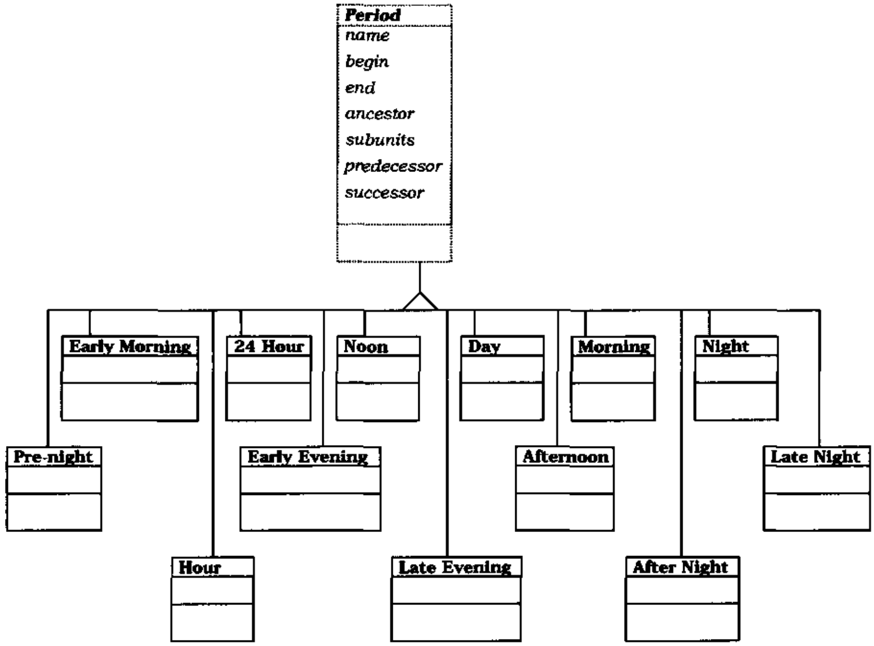


Figure 7-7 Time hierarchy classes.

The attributes[174] of the period subclasses are the variables representing the domain concepts. These variables are themselves instances of subclasses of the class **Variable.** Figure 7-8 shows the **Variable** class and its subclasses in detail. The **Variable** class contains attributes that describe a variable object and attributes needed in constraint reasoning. Subclasses of this class determine the type of a domain concept, which is either numerical or symbolic.

---

not used to generate objects from. 'Normal' classes (*i.e.* classes from which objects can be generated) are shown in roman. Triangle denotes subclass relation.

[174] In the implementational sense, that is: "slot" for Common Lisp, or "data-member" for C^{++}.

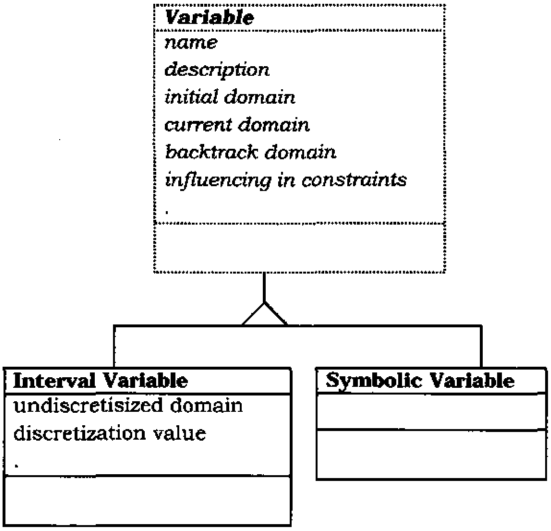


Figure 7-8 Variable classes.

Variables which do not fit in the datastructure presented above (*e.g.* the area of the greenhouse or the number of stems per m^2), belong to the more or less fixed characteristics during a cropping season. They may be stored as attributes belonging to the class **Greenhouse** (or **Compartment**) or in an extension of the current time hierarchy (*e.g.* the following 'period' classes: **Week**[175], **FourWeekPeriod** and **Season** may be added). However, there are likely to remain one or more classes that represent the permanent greenhouse or compartment characteristics.

7.4.3 Constraint generation

After generation of a new branch in the time hierarchy, constraints applicable to the variables of period objects need to be created. Constraints are generated on the basis of constraint templates. A constraint template contains all the necessary information to create constraint objects. *Figure 7-9* shows an example of a constraint template. If this template were to be called by an object of type **Morning**, a constraint would be generated which relates the temperature of the morning object to the temperature of the hour objects that are part of that particular morning object using the `average` relation.

---

[175] Especially the **Week** class is useful for settings of a grower that apply to several days (*e.g.* average three day temperature).

constraint_template `averageTemperature(`*period*`)` {

- description: `"Relates temperature of` *period* `to its subunit temperatures using the average function.";`
- valid_periods: `week, 24-hour, day, night, afterNight, preNight, lateNight, earlyMorning, morning, noon, afternoon, earlyEvening, lateEvening;`
- constraint_function: `average();`
- variables: *period*`.temperature, allSubunits(`*period*`, subunit.temperature); }`

Figure 7-9 Example of a constraint template to create an averageTemperature constraint for the period 'period'.

In Figure 7-9 it can be seen that the attributes of the **Period** class (shown in Figure 7-7) support easy constraint construction, since subunits, ancestors, predecessors, etc. can easily be found.

7.4.4 Constraint classes

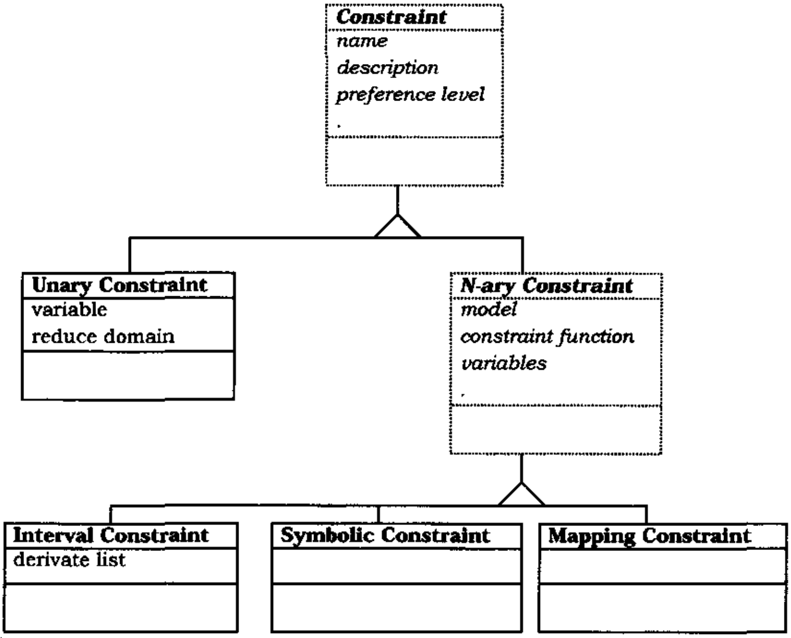


Figure 7-10 Constraint classes.

The constraint reasoning procedures discussed in section 7.2.3 work on individual constraints or on sets of constraints. Using object-oriented techniques it is possible to design procedures that work differently for the specific constraint classes. This different behavior particularly applies to the `revise` procedure of the consistency checking algorithm. Figure 7-10 shows the constraint classes identified here.

Unary constraints
The **UnaryConstraint** class is used to implement the input (*i.e.* decisions, measurements and predictions) to the constraint network. Unary constraints need to be evaluated only once.

Symbolic constraints
The **SymbolicConstraint** class is used to implement qualitative knowledge relations that are part of the qualitative models.

Interval constraints
The **IntervalConstraint** class is used to implement numerical models. In the computations outward-rounded interval arithmetic is used. The `revise` procedure for numerical constraints has been implemented using the interval Newton method (Hansen, 1992).

Mapping constraints
The **MappingConstraint** class is used to link symbolic variables with their numerical counterparts, both variables represent the same concept. These constraints constitute a bridge between qualitative and qualitative knowledge. Mapping constraints are always binary.

7.4.5 *Limitations*

The present prototype of inference engine only implements basic constraint reasoning functionality. The prototype contains a few 'high level' constraint classes and only one basic search algorithm has been implemented[176].

Since the main interest of this work is in the design and modelling of the problem, efficiency has not been treated as a critical issue. Therefore, no procedures have been implemented that allow for further optimization of the efficiency of the constraint reasoning engine. Preprocessing algorithms like clustering constraints into sets that have strong coupling (*e.g.* Dechter and Pearl,1989), and/or grouping constraints in sets that share specific characteristics and applying specific algorithms to these sets (*e.g.* Puget and Leconte, 1995; Van Hentenryck, *et al.*, 1997b) may improve the efficiency of the inference engine significantly. The object-oriented nature

---

[176] For the sake of comparison: commercial constraint reasoning implementations like ILOG SOLVER and COSITEC's CHIP contain elaborate constraint hierarchies *e.g.* CHIP contains more than 100 constraint classes (Simonis and Beldiceanu, 1995). For efficiency reasons, elementary mathematical operations and comparisons like: '+' and '>' have been implemented as separate constraint classes. Such strategy allows one to make adequate use of the specifics of the constraint relation and apply efficient algorithms.

of the implementation in principle allows for easy extension of the constraint hierarchy if this is needed later. However, the above types of improvement will likely require a rearrangement of the present implementation structure and the introduction of new functions and classes (*e.g.* classes representing constraint sets with particular properties).

7.4.6 Use of the prototype in the overall design

The present prototype of the inference engine can be used in the overall design of the supervisory control system in various ways. In a straightforward configuration the scheduler notifies the inference engine that new unary constraints are available. These constraints could imply new measurements for subsequent time intervals, changes in the settings of the grower, an updated weather prediction, or a combination of these. Based on the new constraints, the inference engine generates a series of constraint reasoning problems for each preference level and produces for a solution that has the highest preference value. The prototype can likewise be used for scenario studies since the difference between 'normal' application and scenario studies lays foremost in the data to which the constraint reasoning procedures are applied.

7.5 Particularities of the constraint reasoning framework

7.5.1 Contra-causal domain reduction

Constraint reasoning is a declarative inference method which enforces consistency upon every variable in the constraint network. In constraints that include controllable and external or uncontrollable variables, consistency enforcing implies that the domains of the latter types of variables (which are in practice not influenceable) may also be reduced. The meaning of such reduction must be derived from the concept which the constraint describes and the domains of the other variables in the constraint. For instance consider the model in eqn.(3) for the hourly photosynthesis (Nederhoff, 1994).

$$P = -0.78 + 11.9 \times \frac{I_0}{577 + I_0} \times \frac{CO_2}{221 + CO_2} + 0.18 \times LAI \tag{3}$$

If the objective is to have at least x $mg{\cdot}m^{-2}{\cdot}h^{-1}$ for P, and given that: CO_2 may at most be y ppm, I_0 has originally been predicted to be in $[z_1, z_2]$ $W{\cdot}m^{-2}$, and the LAI is in $[u_1, u_2]$ $m^2{\cdot}m^{-2}$ (this latter interval being the best estimate of the LAI available). After constraint reasoning the lower bound on I_0 has been raised to z_1'. In practice such reduction indicates that the combination of the present state of the crop, the decisions of the grower, and the predicted external conditions (the latter are represented through the interval $[z_1, z_2]$) are only consistent if the actual value will be located inside the reduced interval $[z_1', z_2]$. It is clear that this condition may not be satisfied in reality, hence, it is possible that the settings of the grower will not be realized (or are not appropriate).

A similar argument can be made for constraints that include both fast and slow changing variables, for instance, reduction of the domain of the 'slow' variable *LAI* on the basis of the domain of the 'fast' variable *P* in eqn.(3). In this constraint the domain of the *LAI* is considered non-reducible (in fact, it is yesterday's *LAI*). However, unlike in the previous situation it is not possible to confirm whether there is a real inconsistency since our rough knowledge of the *LAI* does not allow for such a conclusion (unless additional measurements are taken, which is not assumed).

In both cases the treatment will be the same, domain reduction is allowed, that is, consistency is assumed until a 'real' inconsistency can be observed (*i.e.* empty domain for some variable).

7.5.2 Multiple solutions

The implementation of the prototype uses local consistency enforcing combined with search to produce a solution at a given preference level. Such a solution satisfies the settings of the grower and denotes a value-assignment for every variable in the constraint reasoning problem. Once one solution at a given preference level has been found, there could be alternative solutions that also satisfy the grower's settings. If alternative solutions are found, a choice must be made concerning what or which one(s) to send to the setting calculation mechanisms.

Table 7-2 shows an example of such situation, it displays four compound labels that are part of four solutions generated by the inference engine (the discretization value for temperature and CO_2-concentation is: 0.1°C, resp. 10 ppm). Each of them could be sent to the setting calculation mechanism. However, considering that the optimization horizon for the setting calculation mechanism is one hour, the values for the variable $\text{Temperature}_{\text{morning}}$ will not be sent to the setting calculation mechanism[177]. Which means that solutions one and two are identical from the viewpoint of the setting calculation mechanism.

---

[177] Since they cannot be used without further processing.

Table 7-2 Individual solutions.

$\text{Temperature}_{\text{morning}}$	$\text{Temperature}_{9:00\text{-}10:00}$	CO_2-concentration$_{9:00\text{-}10:00}$
[20.0 , 20.1]	[19.9 , 20.0]	[420, 430]
[20.1 , 20.2]	[19.9 , 20.0]	[420, 430]
[20.0 , 20.1]	[20.0 , 20.1]	[420, 430]
[20.0 , 20.1]	[19.9 , 20.0]	[430, 440]

There are at least three methods that could be used to deal with multiple solutions. A simple method – from the viewpoint of the inference engine – is to send all generated solutions to the setting calculation mechanism and have the setting calculation mechanism find devices settings that satisfy one solution. This method requires that the setting calculation mechanism can deal with multiple solutions, but has the advantage that the setting calculation mechanism has the maximum solution space to find combinations of devices settings in.

Another possibility is to have the inference engine select one solution based on an additional optimization criterion. For instance, it could select the compound label that satisfies the highest number of solutions. For the compound label containing the variables $\text{Temperature}_{9:00\text{-}10:00}$, CO_2-concentration$_{9:00\text{-}10:00}$, the combination <[19.9 , 20.0], [420, 430]> would thus be preferred to the other possibilities in Table 7-2. A disadvantage of this method is that the setting calculation mechanism can only calculate devices settings within the limits of the selected solution.

A third possibility is to have the inference engine combine the solutions it has generated. A central issue in combining solutions into solution intervals (for numerical variables) is not to avoid introducing invalid regions. For instance, a simple join of the values for the variables $\text{Temperature}_{9:00\text{-}10:00}$, CO_2-concentration$_{9:00\text{-}10:00}$ in Table 7-2 results in the regions [19.9 , 20.1] and [420, 440]. However, the compound label <[20.0 , 20.1], [430, 440]> for $\text{Temperature}_{9:00\text{-}10:00}$, respectively the CO_2-concentration$_{9:00\text{-}10:00}$, is invalid. In constraint reasoning literature several methods have been presented to deal with this problem. One possibility is to use path consistency algorithms (*e.g.* Tsang, 1993), another possibility is to enforce bounds-consistency (*e.g.* Van Hentenryck *et al.*, 1997b) on the interval variables. Unfortunately, both methods have their own drawbacks, the latter method requires the constraints to be monotonic over the domains of the variables in the constraint, while the former is computationally intensive if the constraint network is not tree-shaped. In the considered problem neither condition applies, therefore these methods will not be used.

The advantages and disadvantages discussed above show that none of the methods has a clear advantage over the others. In the remainder the first method will be assumed, since it is, given the objectives discussed earlier, the easier to apply.

7.5.3 Uncertainty issues

In this section two types of uncertainty are discussed, namely uncertainty with respect to the actual state of the greenhouse–crop system and uncertainty with respect to weather predictions. Both types of uncertainty are responsible for generating additional solutions compared to cases in which uncertainty is not present.

Uncertainty in model relations together with uncertainty coming from (initial) measurements is inherent of the problem domain. This kind of uncertainty can be represented as intervals, where the width of the interval conveys the amount of uncertainty. Alternative solutions that have different interval values for this type of uncertain variables are equally valid and there is no means to determine which solution is best. However, since the setting calculation mechanism only uses a subset of the variables in a solution (*i.e.* a compound label), some compound labels may refer to multiple (complete) solutions. These compound labels may cover a larger section of the domain of the uncertain variable and may therefore be preferred over other compound labels which do not cover multiple solutions.

Uncertainty with respect to weather predictions can be represented as intervals on the variables that denote future external conditions. Alternative solutions with different interval values for these variables refer to alternative external situations[178]. Since only one of these alternative external situations will actually occur, the setting calculation mechanism can use the most recent measurements to determine the solution that agrees best with the actual situation.

In the next chapter, uncertainty will simply be represented as intervals over the variables that bear either type of uncertainty. Of course, one could conceive more elaborate ways to represent uncertainty *e.g.* using probability distributions over the interval itself (*e.g.* Fargier *et al.*, 1994). However, applying such concept would make the present constraint reasoning framework considerably more complex and may therefore be considered a next step in the development cycle of the prototype, if such is considered needed and if the resulting algorithms are sufficiently efficient.

---

[178] The external conditions can be defined as assignments of the external variables I_0, $T_{outside}$, Windspeed and $Rh_{outside}$, over some successive periods.

8. ILLUSTRATING THE FUNCTIONING OF THE INFERENCE ENGINE

8.1 Introduction

In this chapter the behavior of the inference engine will be explained through examples. The objective of this chapter is to show and explain the workings of the prototype and to provide the reader with a feeling for the inference behavior of constraint reasoning algorithms in general and the constraint solver explained in chapter seven in particular.

The examples show the behavior of the inference engine under conditions that simulate its use within the setting of the overall supervisory control system. Each example illustrates a particular aspect of the inference engine's behavior. The following aspects are shown:

- *The workings of the preference distribution.*
  This example shows the overall process of finding solution at a certain preference level on a simple numerical constraint. The construction of constraint reasoning problems according to the preference input of the grower will be shown.
- *The behavior of the inference engine given uncertain data.*
  Based on the previous example the consequences of introducing uncertain data will be explained.
- *The use of cumulative and average constraints in the time hierarchy.*
  Cumulative and average constraints allow the grower to specify longer term objectives and allow for compensation over various periods. The way these constraints influence the solutions generated by the inference engine will be shown here.
- *The combination of qualitative and quantitative constraints.*
  Model and input constraints can be qualitative and quantitative, it will be shown how qualitative and quantitative constraints influence each other.

This chapter will be concluded with a discussion on the integration of the above.

8.2 Procedures for constraint reasoning

The examples are carried out by using calls to the following functions of the inference engine: `generate-constraint-problems`, `revise`, `ac-3-solo` and `forward-checking`. The function `generate-constraint-problems` uses the preference levels to generate constraint problems for every preference level. The `revise` procedure is called on a single constraint and realizes local consistency of the variables in a single constraint. `Ac-3-solo` is called with a list of constraints as its argument, it uses `revise`, and realizes local consistency for all variables that take part in or can be linked to the constraints in the list. Since each variable contains a reference to the list of constraints in which it is involved (Figure 7-2), a call to `ac-3-solo` on a list containing only one constraint can lead to a complete update of the constraint network. Finally, the `forward-checking` procedure finds all solutions at a given preference level.

8.3 Preference distribution

In this example the workings of the preference distribution will be shown on a small system of constraints. This system contains the following constraints: a constraint describing the hourly photosynthesis for tomato (eqn. (4), model according to Nederhoff, 1994), two constraints that represent the settings of the grower (Figure 8-1), and a constraint (I_0 = 300 W·m^{-2}) that expresses the relevant external conditions (the light intensity). The value for the variable *LAI* (leaf area index) is set to 3 m^2·m^{-2}, the discretization values for the variables CO_2 and *Photosynthesis* are respectively 25 ppm and 0.1 g·m^{-2}·h^{-1}.

$$c_1 : P = -0.78 + 11.9 \times \frac{I_0}{577 + I_0} \times \frac{CO_2}{221 + CO_2} + 0.18 \times LAI \qquad (4)$$

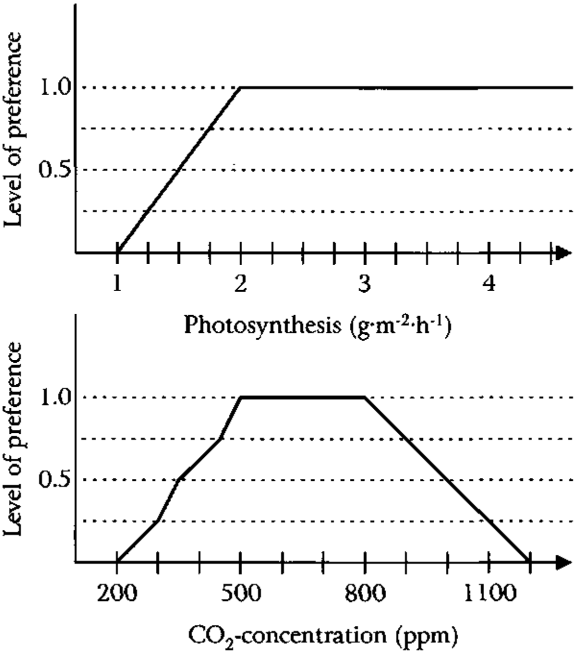


Figure 8-1 Preference distribution of the grower for photosynthesis and the CO$_2$-concentration.

As can be seen from the preference distributions in Figure 8-1, the variable *Photosynthesis* has a lower bound and the variable CO_2 has both a lower and an upper bound. The preference distribution for the variable *Photosynthesis* indicates that the grower prefers higher *Photosynthesis* levels to lower ones. For the variable CO_2, the grower has distinct ranges that he prefers to other ranges. The rationale for an upper bound for the CO_2-concentration is that high levels of CO_2 can – under specific condition – damage the crop (additionally, cost considerations may apply as well).

The output of the inference engine is a set of solutions for the variables that are available in the setting calculation mechanism. In this example it is assumed that only the variable CO_2 is available in the setting calculation mechanism.

When the above input is processed by the function `generate-constraint-problems`, it results in five constraint reasoning problems. They are shown in Table 8-1, the domains for the variables *Photosynthesis* and CO_2 denote their initial intervals, that is, before processing by the constraint reasoning engine.

Table 8-1 Five constraint reasoning problems.

Preference	*Photosynthesis*	CO_2	I_0	*LAI*
0	> 1	[200, 1200]	300	3
0.25	> 1.25	[300, 1100]	300	3
0.5	> 1.5	[350, 1000]	300	3
0.75	> 1.75	[450, 900]	300	3
1	> 2	[500, 800]	300	3

Table 8-2 shows the results after constraint reasoning. The domains for the variables *Photosynthesis* and CO_2 denote their domains after filtering, the number of solutions denotes the size of the solution set after applying the `forward-checking` procedure. As can be seen from this table, the highest preference level for which a solution can be found is 0.5. At levels 0.75 and 1, the demands of the grower are incompatible with the current crop state (represented by the *LAI*) and the prevailing external conditions (represented by the variable I_0). One may also note in Table 8-2 that the lower bound for CO_2 after consistency enforcing is much higher that the bound given by grower. This is due to the constraints on the variables *Photosynthesis* and I_0 (given the model and the value for the *LAI*), therefore, in this case the lower bound was redundant.

Table 8-2 Solutions to the five constraint reasoning problems.

Preference	*Photosynthesis*	CO_2	No of solutions
0	[1.00, 1.83]	[225, 1200]	42
0.25	[1.25, 1.80]	[341, 1100]	34
0.5	[1.50, 1.77]	[538, 1000]	19
0.75	-	-	-
1	-	-	-

8.4 Uncertainty

The second example shows the consequences of uncertainty in the above setting. Various kind of uncertainty could be introduced. For instance, the actual state of the crop, represented by the variable *LAI*, could be imprecise. Instead of using the precise value 3, an interval, *e.g.* [2.8, 3.2], could be used. Second, the prediction of the external conditions could incorporate some uncertainty, instead of using the value 300 W·m^{-2} for the variable I_0, an interval, for instance [270, 330] W·m^{-2}, could be used. Finally, the parameters in the models can be imprecise. Since eqn. (4) is a regression model for the photosynthesis, its parameters necessarily include some uncertainty. Instead of using average parameter values, confidence intervals can be used. Since the prototype applies interval arithmetic, the use of confidence intervals instead of average parameter values does not require additional effort[179].

Table 8-3 Parameter values and their 95% confidence interval for tomato (Nederhoff, 1994: p.33-34).

Average parameter value (st.dev.)	Interval value (95% confidence, n=1568)
-0.78 (0)	[-0.78, -0.78]
11.9 (0.6)	[11.87,11.93]
577 (42)	[574.9 579.1]
221 (14)	[220.3 221.7]
0.18 (0.02)	[0.179 0.181]

---

[179] That is, no additional effort is required during constraint definition. During constraint reasoning, interval arithmetic is computationally more expensive (on of the shelf hardware) than straight forward floating point arithmetic.

For instance, using the confidence intervals of Table 8-3 in eqn. (4) instead of the average values on the tuple <*LAI*=3, I_0=300 and CO_2=500>, evaluates to the interval [1.449, 1.475]. Using the equation with average parameter values results in the value 1.463 (given the same input). A difference of ±1.0%. When these confidence intervals are used one can be 95% confident that the interval actually contains the "true" value for the variable *Photosynthesis*.

In the following two cases the other types of uncertainty are explored. Unless stated otherwise, the constraints of section 8.3 are applicable.

In the first case, the influence of uncertainty in the crop state (*LAI*=[2.8, 3.2]) will be shown. Table 8-4 shows the results after constraint reasoning. For every preference level, the domains (after filtering) for the variables *Photosynthesis*, CO_2 and *LAI* are shown, the number of solutions denote the size of the solution set after applying the `forward-checking` procedure. Compared to the results in Table 8-2 one notices the following differences. First, solutions are found at a higher preference level than in the previous case, that is, if (and only if) the *LAI* is higher than 3.1, two solutions at the 0.75 preference level exist. Of course, whether the *LAI* is actually higher than this value can only be found out *after* additional measurements, therefore without these measurements one cannot tell if this solution is really valid. Second, the `forward-checking` procedure has found more solutions at higher preference levels. The increased width of the *LAI* causes more instantiations of CO_2 and *Photosynthesis* to conform to the constraints (the width of the domains after filtering indicates this), however some of these solutions will also exhibit a reduction of the *LAI*.

In the second case, the influence of uncertainty in the external conditions (I_0) will be shown. Table 8-5 shows the results after constraint reasoning. For every preference level, the domains (after filtering) for the variables *Photosynthesis*, CO_2 and I_0 are shown (*LAI*=3.0). The number of solutions in Table 8-5 denotes the size of the solution set after applying the `forward-checking` procedure. Compared to the results in Table 8-2 and Table 8-4 one notices that at every level (except the 1.0 level) the number of solutions are much higher. This increase in the size of the solution space is due to discretization of the variable I_0. Unlike the *LAI*, this variable has a discretization value (λ = 20 W·m^{-2}). The discretization value of the light intensity causes the constraint reasoning engine to generate solutions for alternative

Table 8-4 Inference results with respect to the uncertainty in the variable LAI.

Preference	*Photosynthesis*	CO_2	*LAI*	No of solutions
0	[1.00, 1.87]	[213, 1200]	[2.8, 3.2]	71
0.25	[1.25, 1.84]	[320, 1100]	[2.8, 3.2]	54
0.5	[1.50, 1.81]	[501, 1000]	[2.8, 3.2]	31
0.75	[1.75, 1.77]	[861, 900]	[3.1, 3.2]	2
1	-	-	-	-

Table 8-5 Inference results with respect to the uncertainty in the variable I_0.

Preference	*Photosynthesis*	CO_2	I_0	No of solutions
0	[1.00, 1.94]	[204, 1200]	[280, 320]	161
0.25	[1.25, 1.91]	[301, 1100]	[280, 320]	126
0.5	[1.50, 1.88]	[456, 1000]	[280, 320]	76
0.75	[1.75, 1.83]	[741, 900]	[304, 320]	7
1	-	-	-	-

external conditions[180]. Like in Table 8-4 it can be observed that solutions at the 0.75 preference level exist only for particular external conditions.

When the amount of uncertainty in the variable I_0 is increased even further, the number of solutions increases correspondingly. This is shown in Table 8-6.

Table 8-6 Inference results after further increase of the uncertainty in the variable I_0.

Preference	*Photosynthesis*	CO_2	I_0	No of solutions
0	[1.00, 1.99]	[200, 1200]	[270, 330]	241
0.25	[1.25, 1.96]	[300, 1100]	[270, 330]	185
0.5	[1.50, 1.93]	[424, 1000]	[270, 330]	113
0.75	[1.75, 1.88]	[670, 900]	[304, 330]	20
1	-	-	-	-

8.5 Time hierarchy

In this example the behavior of the inference engine will be shown in the context of a small time hierarchy. The objective is to show its behavior while simulating a sequence of events that could occur in the greenhouse. In this example no preference distribution will be used since it will not add much to the purpose of the example.

---

[180] The setting calculation mechanism can (at the last moment) choose a solution set that belongs to the alternative external condition that fits the actual situation best.

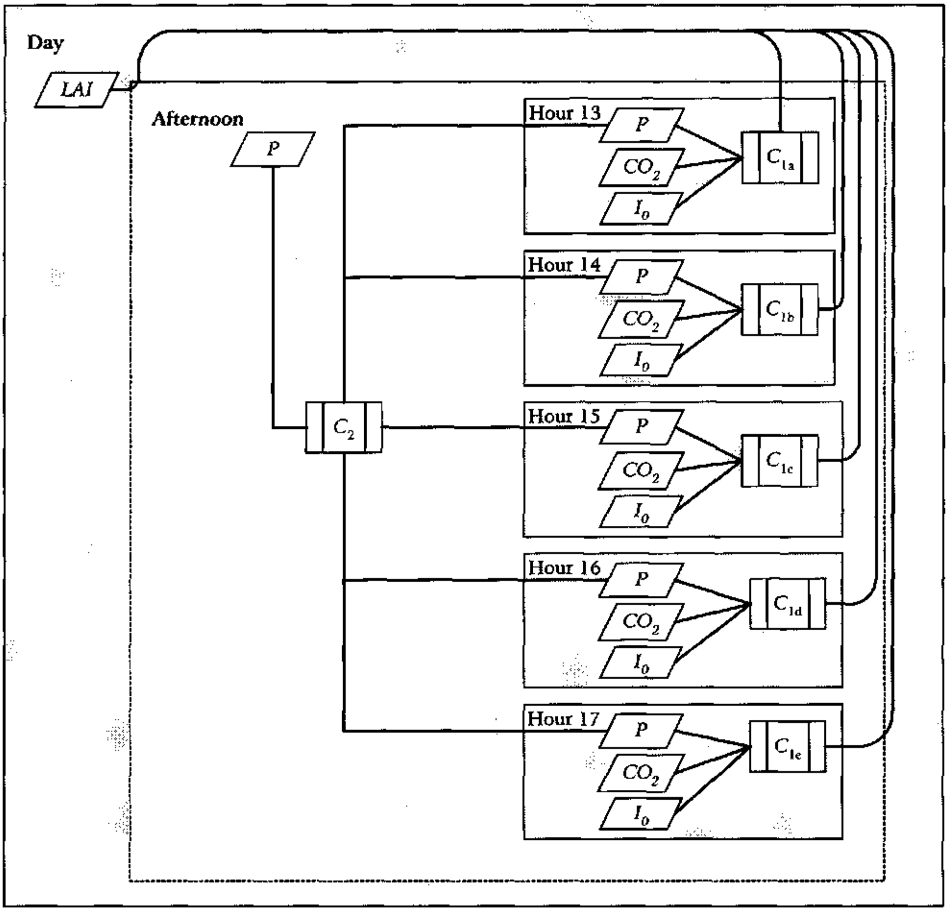


Figure 8-2 The time hierarchy.

The time hierarchy is shown in Figure 8-2, it contains three levels: the day, the afternoon and the hour level. Furthermore, it contains six constraints: C_{1a}...C_{1e} and C_2 (eqn.'s (4) resp. (5)), and a small number of variables (*LAI*, *P*, and {*P*, CO_2, I_0} at their respective levels).

$$C_2 : P_{afternoon} = \sum_{hour=13}^{17} P_{hour} \tag{5}$$

The constraint C_2 in eqn.(5) describes the accumulated photosynthesis (in g·m^{-2}) for the afternoon and involves five occurrences of P_{hour}. This constraint is special in the sense that it in principle allows for compensating behavior. When the grower sets a minimum bound on $P_{afternoon}$, this constraint can be reached through alternative combinations of P_{13}...P_{17}, for instance, high values for P_{13}...P_{15} can compensate for low values for $P_{16}...P_{17}$ (or some other combination) and thus satisfy the bound set by the grower.

To investigate the behavior of the inference engine, Table 8-7 through Table 8-15 show the results after applying the constraint solver on a number of examples. The constraint reasoning problems consist of: *i*) a sequence of estimated, resp. externally constrained and/or measured values for the variables $I_{0,\ hour}$ and $CO_{2,\ hour}$, and *ii*) a unary constraint on $P_{\text{afternoon}}$, that simulates an objective of the grower. The discretisation values for CO_2, I_0 and P_{hour} are resp. 25 ppm, 20 Wm^{-2} and 0.125 $g{\cdot}m^{-2}{\cdot}h^{-1}$.

Table 8-7 shows the initial situation: *i*) intervals for I_0 for the next five hours (obtained through some kind of prediction), *ii*) the next five hourly intervals for CO_2 (it is assumed that they are constrained by objectives that are not included in the example), *iii*) the initial domains for P_{hour}, and, *iv*) the constraint $P_{\text{afternoon}} > 7.5$ $g{\cdot}m^{-2}$, simulating an objective of the grower, that is, his preferred domain for the variable "afternoon photosynthesis". The *LAI* is 3 $m^2{\cdot}m^{-2}$.

Table 8-7 Initial situation.

time instances	$I_{0,\ hour}$	$CO_{2,\ hour}$	P_{hour}	$P_{afternoon}$
13	[360, 370]	[470, 480]	[0, 8]	
14	[400, 430]	[400, 450]	[0, 8]	
15	[350, 400]	[450, 500]	[0, 8]	
16	[300, 350]	[450, 520]	[0, 8]	
17	[250, 350]	[450, 550]	[0, 8]	
afternoon				> 7.5

Table 8-8 shows the domains of the variables after applying the `ac-3-solo` procedure. In the table caption, the number of solutions generated by the `forward-checking` procedure and its runtime are shown. From this table, it can be seen that the constraint on $P_{\text{afternoon}}$ has no influence because the present lower bound of $P_{\text{afternoon}}$ is higher than the lower bound given in the constraint.

Table 8-8 Variable domains after applying constraint reasoning, no. of solutions: 4320, runtime[181]*: 654 sec.*

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	[360, 370]	[470, 480]	[1.68, 1.74]	
14	[400, 430]	[400, 450]	[1.73, 1.93]	
15	[350, 400]	[450, 500]	[1.62, 1.88]	
16	[300, 350]	[450, 520]	[1.41, 1.70]	
17	[250, 350]	[450, 550]	[1.18, 1.74]	
afternoon				[7.62, 8.99]

8.5.1 An increase in information over time

Table 8-9 to Table 8-12 simulates successive situations in which more information becomes gradually available. That is, every hour estimated values become more precise, measured values for past time steps will be obtained, and as a consequence, the domain for $P_{afternoon}$ becomes more narrow (after filtering), moreover, the number of possible solutions falls dramatically.

Table 8-9 Variable domains after applying constraint reasoning, no. of solutions: 144, runtime: 28.3 sec.

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	*[365, 365]*	*[475, 475]*	*[1.71, 1.71]*	
14	[420, 430]	[400, 420]	[1.81, 1.88]	
15	[370, 400]	[450, 500]	[1.70, 1.88]	
16	[300, 350]	[450, 500]	[1.41, 1.68]	
17	[250, 300]	[450, 500]	[1.18, 1.46]	
afternoon				[7.80, 8.61]

---

[181] Pentium 133MHz, 32Mb ram, Allegro Common Lisp version 3.0 (Franz Inc., Berkeley (CA), USA).

Table 8-10 Variable domains after applying constraint reasoning, no. of solutions: 24, runtime: 4.7 sec.

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	*[365, 365]*	*[475, 475]*	*[1.71, 1.71]*	
14	*[430, 430]*	*[400, 400]*	*[1.84, 1.84]*	
15	[390, 400]	[450, 470]	[1.78, 1.84]	
16	[300, 325]	[450, 500]	[1.41, 1.57]	
17	[250, 300]	[450, 500]	[1.18, 1.46]	
afternoon				[7.92, 8.43]

Table 8-11 Variable domains after applying constraint reasoning, no. of solutions: 6, runtime: 1.4 sec.

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	*[365, 365]*	*[475, 475]*	*[1.71, 1.71]*	
14	*[430, 430]*	*[400, 400]*	*[1.84, 1.84]*	
15	*[390, 390]*	*[470, 470]*	*[1.80, 1.80]*	
16	[300, 310]	[480, 500]	[1.44, 1.51]	
17	[250, 300]	[450, 500]	[1.18, 1.46]	
afternoon				[7.98, 8.33]

Table 8-12 Variable domains after applying constraint reasoning, no. of solutions: 2, runtime: 0.7 sec.

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	*[365, 365]*	*[475, 475]*	*[1.71, 1.71]*	
14	*[430, 430]*	*[400, 400]*	*[1.84, 1.84]*	
15	*[390, 390]*	*[470, 470]*	*[1.80, 1.80]*	
16	*[300, 300]*	*[490, 490]*	*[1.45, 1.45]*	
17	[250, 270]	[470, 500]	[1.20, 1.32]	
afternoon				[8.01, 8.13]

8.5.2 The influence of constraint tightening

Starting from the initial situation (Table 8-7), the constraint on $P_{afternoon}$ has been tightened by increasing the lower bound from 7.5 to 8.25 $g{\cdot}m^{-2}$. Table 8-13 shows the results. Tightening the constraint on $P_{afternoon}$ has no effect on the domains of P_{13}

to P_{17} (and $I_{0,\ 13...17}$ and $CO_{2,13...17}$) but reduces the number of solutions (compare Table 8-13 with Table 8-8).

Table 8-13 Variable domains after applying constraint reasoning, no. of solutions: 4300.

time instances	$I_{0,\ hour}$	$CO_{2,\ hour}$	P_{hour}	$P_{afternoon}$
13	[360, 370]	[470, 480]	[1.68, 1.74]	
14	[400, 430]	[400, 450]	[1.73, 1.93]	
15	[350, 400]	[450, 500]	[1.62, 1.88]	
16	[300, 350]	[450, 520]	[1.41, 1.70]	
17	[250, 350]	[450, 550]	[1.18, 1.74]	
afternoon				[8.25, 8.99]

No change in the domains of P_{13} to P_{17} together with a reduced number of solutions[182] means that some of the compound labels that can be generated from the filtered domains of P_{13} to P_{17} are inconsistent. Hence, local consistency does not imply global consistency and some parts of the domains of P_{13} to P_{17} in combination are not valid. This is shown in Table 8-14, where the $CO_{2,\ 13}$ and $I_{0,\ 13}$ have been instantiated at their lower bounds and the upper bounds of $CO_{2,\ 14}$, $I_{0,\ 14}$, $I_{0,\ 15}$ and $CO_{2,\ 17}$ have been lowered. These changes result in an increase in the lower bound of variables $I_{0,\ 16}$ and $I_{0,\ 17}$, from 300 to 306 $W{\cdot}m^{-2}$, resp. from 250 to 259 $W{\cdot}m^{-2}$. Conceptually, such an increase indicates that only if the light intensity is higher than this lower bound the objective $P_{afternoon} \geq 8.25\ g{\cdot}m^{-2}$ can be realized. Since the predictions for $I_{0,\ 16}$ and $I_{0,\ 17}$ are [300, 350] resp. [250, 300], this is not sure.

Table 8-14 Variable domains after applying constraint reasoning, no. of solutions: 82.

time instances	$I_{0,\ hour}$	$CO_{2,\ hour}$	P_{hour}	$P_{afternoon}$
13	[360, 360]	[470, 470]	[1.68, 1.68]	
14	[400, 410]	[400, 420]	[1.73, 1.80]	
15	[350, 375]	[450, 500]	[1.62, 1.78]	
16	[307, 350]	[450, 520]	[1.51, 1.70]	
17	[259, 350]	[450, 500]	[1.27, 1.46]	
afternoon				[8.25, 8.44]

---

[182] Provided that other influencing factors (*e.g.* variable ordering) remain the same.

Again, starting from the initial situation (Table 8-7), the constraint on $P_{\text{afternoon}}$ has been further tightened by increasing the lower bound to 8.75 g·m^{-2}. Table 8-15 shows the results.

Table 8-15 Variable domains after applying constraint reasoning, no. of solutions: 657.

time instances	$I_{0,\,hour}$	$CO_{2,\,hour}$	P_{hour}	$P_{afternoon}$
13	[360, 370]	[470, 480]	[1.68, 1.74]	
14	[400, 430]	[400, 450]	[1.73, 1.93]	
15	[350, 400]	[450, 500]	[1.65, 1.88]	
16	[300, 350]	[450, 520]	[1.47, 1.70]	
17	[297, 350]	[450, 550]	[1.50, 1.74]	
afternoon				[8.75, 8.99]

Although the number of solutions has decreased substantially (from 4300 to 657 solutions), Table 8-15 shows that only P_{15}, P_{16}, P_{17} and $I_{0,\,17}$ are effected by this very tight bound. This behavior is caused by the possibilities for compensation among the individual time-instances.

8.6 Combining qualitative and quantitative knowledge

This section illustrates mixing qualitative and quantitative variables in a qualitative–quantitative constraint network. It will be shown how qualitative and quantitative variables interact through the combined use of qualitative, quantitative and mapping constraints. However before doing so the behavior of the inference engine on qualitative constraints will be discussed first.

8.6.1 Qualitative constraints

In this example constraints with variables that have qualitative or symbolic domains are used. The example shows the operation of the constraint reasoning procedures on the qualitative model of the risk of getting Blossom-end rot (BER) presented in appendix one. This model relates a “risk of getting BER” to other crop processes and environmental conditions in the greenhouse. With this model a grower can strive for acceptable levels on the "risk of getting BER", because given a present risk level and the risk level aimed at, the grower can investigate the conditions that - according to this model - realize his objective.

As input constraints for this small constraint network, reduced domains for the variables calcium demand (Calcium-Demand-Day-1), daytime transpiration (Day-

Table 8-16 The initial domains, the domains after filtering and an example solution for the variables in the BER model.

variable	initial domain	filtered domain	example solution
Ber-Risk-Day-1	(VL L ML AV)	(AV)	(AV)
Ber-Risk-Day-0	(MH)	(MH)	(MH)
Ber-Risk-Change-Day-1	(VL L ML AV MH H VH)	(ML)	(ML)
Calcium-Demand-Day-1	(AV MH H VH)	(AV)	(AV)
Calcium-Supply-Day-1	(VL L ML AV MH H VH)	(MH)	(MH)
Calcium-Uptake-Day-1	(AV)	(AV)	(AV)
Nighttime-Ca-Supply-Day-1	(VL L ML AV MH H VH)	(MH)	(MH)
Nighttime-Transp-Day-1	(VL L ML AV MH H VH)	(ML AV)	(AV)
Daytime-Transp-Day-1	(AV MH H VH)	(AV MH H VH)	(H)
Daytime-Ca-Supply-Day-1	(VL L ML AV MH H VH)	(VL L ML AV)	(L)

time-Transp-Day-1), and the calcium uptake (Calcium-Uptake-Day-1) are given[183]. The filtered domains for these variables are used to further reduce the number of solutions that realize the objective. In this example, the preceding day resulted in a risk of getting BER (Ber-Risk-Day-0) of "moderately high" (MH) while the grower would like to reduce this risk to at least "average" (AV) for the present day (Ber-Risk-Day-1). The calcium demand is estimated to be at least "average", which is also the case for the estimated day-time transpiration. The calcium uptake is expected to be "average".

Given these premises, can a solution be found? As can be seen in the third column of Table 8-16 solutions may be possible since the domain of every value after applying the `ac-3-solo` procedure contains at least one value. After applying the `forward-checking` procedure eight solutions were found, as an example one of them is shown in the last column of Table 8-16 (although all eight solutions are equally valid). So indeed the simulated objective of the grower can be realized under the given starting conditions. However, if the calcium uptake had been "moderately low" (ML) instead of "average" no solution would have existed.

---

[183] Note that normally, these values are automatically generated through constraint reasoning within the overall constraint network.

Using the BER model[184] the workings of the `revise` method can easily be explained. For instance, given the value for the "risk of getting BER" for the previous day (*i.e.* variable: "Ber-Risk-Day-0" = "MH") and the objective to decrease this value for the pending day to one of the values "very low", "low", "moderately low" and "average" (*i.e.* variable "Ber-Risk-Day-1" ∈ {VL, L, ML, AV}), Table 1 in appendix one can be used to look up the admissible values for the required change in the risk of getting BER (*i.e.* variable: "Ber-Risk-Change-Day-1"). The table shows that the only values admissible for the variable "Ber-Risk-Change-Day-1" are: "very low", "low" and "moderately low". Thus by applying the constraint, the domain of variable "Ber-Risk-Change-Day-1" has been reduced to these three values. Furthermore, this table shows that given these three values for "Ber-Risk-Change-Day-1", the variable "Ber-Risk-Day-1" can only have the value "average" assigned, the values: "very low", "low" and "moderately low" are inadmissible. The `revise` procedure will thus remove these values from the domain of "Ber-Risk-Day-1"; as can be observed in third column of Table 8-16.

8.6.2 Mapping constraints

To combine qualitative and quantitative variables a special constraint type is necessary. This section illustrates the behavior of this constraint type. Mapping constraints project symbolic variables onto their numerical counterparts and vise versa.

Table 8-17 Symbolic - numerical mapping for relative humidity[185] ($x_{symbolic}$ resp. $x_{numerical}$) using a qualitative scale of seven values.

Symbolic value	Numerical value
very low (VL)	[0, 60]
low (L)	[60, 65]
moderately low (ML)	[65, 70]
average (AV)	[70, 75]
moderately high (MH)	[75, 80]
high (H)	[80, 85]
very high (VH)	[85, 100]

---

[184] Table 1 in appendix 1 represents the constraint between the variables: "Ber-Risk-Day-1", "Ber-Risk-Day-0" and "Ber-Risk-Change-Day-1".

[185] The boundaries of the intervals given in this table are only stated for the sake of the example, it is up to the experts (*e.g.* extension specialists and growers) to decide - on the basis of their expert knowledge - what the actual interval boundaries should be.

In constraints of this type, the domains of the variables are reduced using a table (or function) that derives symbolic values from intervals and vise versa. Mapping constraints always work on pairs of variables that represent the same concept, only their representation differs.

As an example, consider the variable $x_{symbolic}$ with a domain: {"very low", "low", moderately low"}, the variable $x_{numerical}$ with domain: [60, 83] and a constraint C_{map} that uses Table 7-1. Applying the `revise` method to the constraint C_{map} will reduce the domain of $x_{symbolic}$ to: {"low", "moderately low"} and that of $x_{numerical}$ to: {[60, 70]}. Because $x_{numerical}$ does not (partly) support the symbolic value "very low" it is excluded from the domain of $x_{symbolic}$, also, the interval [70, 83] is not supported by any of the values of $x_{symbolic}$ and is thus removed.

8.6.3 A mixed qualitative–quantitative constraint network

To be able to use the qualitative constraints that describe "the risk of getting BER" in a mixed qualitative–quantitative constraint network, additional variables and mapping constraints are needed (Figure 2 in appendix one). With respect to the qualitative variables used in section 8.6.1, numerical counterparts are needed for the variables: "Calcium-Uptake-Day-1", "Calcium-Demand-Day-1", "Daytime-Transp-Day-1", and "Daytime-Transp-Day-1".

Table 8-18 Symbolic - numerical mapping for the daytime transpiration (in l per plant)[186].

Daytime-Transp-Day-1 $_{symbolic}$	Daytime-Transp-Day-1 $_{numerical}$
very low (VL)	[0, 0.15]
low (L)	[0.15, 0.3]
moderately low (ML)	[0.3, 0.40]
average (AV)	[0.4, 0.55]
moderately high (MH)	[0.55, 0.65]
high (H)	[0.65, 0.80]
very high (VH)	[0.80, 1.25]

Furthermore, mapping constraints that relate the symbolic variables to their numerical counterparts must also be available. When both quantitative variables and mapping constraints are provided a mixed qualitative–quantitative constraint net-

---

[186] Again, the boundaries of the intervals given in this table are only stated for the sake of the example.

work can be constructed and the results of the "qualitative" inference steps can be propagated to the quantitative part of the overall constraint network (and vice versa).

In an example of such propagation the mapping constraint shown in Table 8-18, and the results for the symbolic variables presented in Table 8-16 are used.

For instance, given an initial domain of [0,2] for Daytime-Transp-Day-$1_{numerical}$ and the domain for Daytime-Transp-Day-$1_{symbolic}$ in the third column of Table 8-16 (the values 'AV' to 'VH'), the reduced domain for Daytime-Transp-Day-$1_{numerical}$ (after applying consistency enforcing) is: [0.4, 1.25]. The domain of the variable Day-time-Transp-Day-$1_{symbolic}$ is not changed through the mapping constraint. The value of Daytime-Transp-Day-$1_{symbolic}$ in the example solution shown in Table 8-16 conforms to the resulting domain [0.65,0.80] for Daytime-Transp-Day-$1_{numerical}$. Depending on the discretization value of the variable Daytime-Transp-Day-$1_{numerical}$ this domain contains one or more unique solutions.

8.7 Concluding remarks

In this chapter the behavior of the procedures: `generate-constraint-problems`, `revise` , `ac-3-solo` and `forward-checking` that make up the inference engine was shown.

The first examples show how the preference distribution can be applied and how uncertainty influences the behavior of the inference engine is. The constraint reasoning framework readily allows using uncertainty through simple intervals, however the examples clearly indicate that too much uncertainty can result in the generation of a large number of solutions.

In the time-based data structure, the example shows that initially, at the start of the period, many solutions remain possible. Gradually, during the course of a day more information becomes available and the search space reduces significantly. Table 8-13 through Table 8-15 show that choices at the start of the period, together with certain predictions for the external variables, determine the amount of "maneuvering room" at later time steps. Some choices, especially values near the bounds of the domains, may eventually lead to inconsistency (at a given preference level) later in the day. In the next chapter a method will be discussed that assists in finding the safest solution.

The examples show that the number of initial solutions is particularly high when: *i*) objective constraints hardly constrain the search space, *ii*) constraints on external variables are loose (much uncertainty), and *iii*) the discretization values of the numerical variables are small. Under these conditions it will be quickly impossible to find all solutions. It is clear that narrow (albeit reliable) predictions of external variables will be necessary. Also, the discretization value of external and control variables should not be too small.

Finally, in section 8.6.3 it was shown that qualitative and quantitative knowledge can be combined whenever appropriate mapping relations are available. Acquiring these mapping constraints will not be a trivial undertaking. For instance, for daytime transpiration, the boundaries for the respective intervals of the qualitative values "very low" to "very high" will likely vary during the course of a season and with the size of the crop. Therefore, the tables that represent these mapping constraints may be subject to regular (possibly partly automatic) changes.

9. EPILOGUE OF PART TWO: DESIGN OF A SOLUTION USING CONSTRAINT REASONING

9.1 Introduction

Now that the operation of the inference engine has been illustrated, its place within the supervisory control system can be discussed more clearly. In this chapter the properties of the inference engine will be further explored in the light of the overall supervisory control system. This discussion will among others lead to a better understanding of how the results of the inference engine can be processed further, and to more insight into the current status and desired improvements of the prototype.

The remainder of this chapter consists of the following. First, some issues w.r.t. problem representation and behavior of the inference engine will be explored. Second, the interface and the interaction between the inference engine and the setting calculation mechanism will be discussed. Third, grower–system interaction will be discussed in the light of the results generated by the inference engine. Finally, some conclusions will be drawn on the above.

9.2 Problem representation and resolution

9.2.1 Size of the constraint reasoning problem in the inference engine

In systems that involve planning (compared to systems that are strictly reactive) the choice of the planning horizon is important. The planning horizon represents a future time interval in which one looks ahead while taking decisions for the current time step. The choice of the planning horizon is an important one because it influences the size and thus the difficulty of the constraint reasoning problem[187]. The size of a constraint reasoning problem depends to a large extent on the number of variables involved in the problem. In our case, the number of variables in the inference engine essentially depends on the number of 'future' time instance objects that have been created. The problem will become very large if many time instance objects are created, therefore unnecessary creation of time instance objects must be avoided as the problem may otherwise become impossible to solve within acceptable time bounds.

When looking at the intrinsic attributes of the problem domain a natural planning horizon can sometimes be identified. In operational crop production management

---

[187] Other factors that influence resolution difficulty are: the domain size, the complexity of the constraint calculations, complexity of the constraint network, etc..

(the remainder of) the growing season would be a natural planning horizon. However, due to the uncertainty in weather and the above mentioned computational difficulties related to large planning horizons, it is impractical to use this horizon. The planning horizon must therefore be chosen based on other logical and practical grounds. In our case a more practical planning horizon can be deduced from the averaging capabilities of the crop (De Koning, 1994) and the accuracy of weather predictions. Both factors point to a planning horizon of three to five days. The time hierarchy should therefore include the next three to five days, thereby allowing the grower to plan his settings for this period.

Generating 4 days in advance, and assuming 20 variables at the hour level, results in approximately 2000 variables (for the hour level alone). Considering this initial number of variables alone, it can be expected that the constraint reasoning problem of the inference engine needs extensive tuning and optimization to realize the required computation time bounds[188]. Tuning may among others involve limiting the size by leaving out the day-branches that lay in the future (*e.g.* day three and four). However, if particular levels in the time-hierarchy are missing, the constraint reasoning problem will be incomplete (not all models can be included because variables are missing). To resolve this situation, it may be possible to use constraints from alternative domain models[189].

At this point, many practicalities regarding the resolution of the full constraint reasoning problem need to be resolved, therefore further exploration of the constraint reasoning framework is needed.

9.2.2 The constraint network

The constraint network in the inference engine relates the variables which interest the grower through the constraint relations that are part of the domain models. A model usually consists of multiple equations and additional variables that only play a role within the model at hand. These intermediate variables must - in the context of constraint reasoning - also be assigned a value (and possibly be discretisized), consequently, they enlarge the constraint reasoning problem (and make it more difficult to solve). Therefore, care should be taken when introducing intermediate variables that have no interest in the eyes of the grower. In the example of chapter

---

[188] In fact, estimating in advance whether a constraint reasoning problem is satisfiable is difficult and, in general, NP-hard (*e.g.* Garey and Johnson, 1979; Hayes, 1997). Furthermore, it is equally difficult to predict the relationship between problem size and speed of resolution for average (*i.e.* non-worst case) case complexity (*e.g.* Gent and Walsh, 1996).

[189] For instance, for the photosynthesis process there also exist model equations that relate the daily photosynthetic production to the daily accumulated global radiation (in $J \cdot m^{-2} \cdot day^{-1}$).

eight this could for instance mean that if the grower is not interested in the hourly photosynthesis but only in the afternoon photosynthesis, the six constraints in the example in section 8.5 could be replaced by the single (albeit more complicated) constraint below:

$$P_{afternoon} = \sum_{i=13}^{17} \left(-0.78 + 11.9 \times \frac{I_{0_i}}{577 + I_{0_i}} \times \frac{CO_{2_i}}{221 + CO_{2_i}} + 0.18 \times LAI \right) \qquad (6)$$

Among others, using this constraint could reduce the number of solutions in the examples in section 8.5 because the variables $\mathrm{Phot}_{\mathrm{hour}=13} \ldots \mathrm{Phot}_{\mathrm{hour}=17}$ need not to be discretisized.

9.2.3 Uncertainty

The previous chapter showed that uncertainty in the predictions has a profound influence on the number of solutions and the speed of calculation. The choice of the amount of uncertainty to take into account in future external situations, requires weighing the consequences of an error in prediction against the response time of the system (because of the computational cost required to process the additional situations).

Additionally, it is possible to introduce probability distributions within the constraint reasoning setting to discriminate between external situations (Fargier *et al*, 1994). Such probability distributions may limit the number of possible external cases (solutions) that need to be computed, however keeping track of the total probability per partial instantiation requires additional computational effort. It is difficult to determine at this time whether using probability distributions will be advantageous therefore, further work is needed.

9.2.4 Algorithms

The choice for constraint reasoning as principal resolution technique for the inference engine is based on its capabilities for problem representation as well as its performance during problem solving. Representation issues like flexibility w.r.t. the preferences of the grower, or imprecision in the actual state of the system causes interval constraint reasoning to be an appealing representation mechanism for the numerical variables in the problem specification[190]. Furthermore, constraint reasoning allows for easy combination of qualitative and quantitative knowledge. With respect to performance, it has been reported (*e.g.* Van Hentenryck *et al.*, 1997a)

---

[190] The modelling of preferences has been given a theoretical support within the framework of possibility theory and the associated approaches to manage flexible constraints (*e.g.* Dubois and Prade, 1986; Fargier *et al.* 1994).

that interval methods within a constraint reasoning framework can perform extremely well.

The algorithms used in the prototype are sufficient to show the application of constraint reasoning on some basic examples. For instance, the choice for the forward checking algorithm has been made based on its simplicity and ease of use. Among others, forward checking allowed for easy combination of qualitative and quantitative constraint relations. However, for augmentation to the full-scale constraint reasoning problem, the algorithms need to be improved, extended or even replaced by more efficient ones (*e.g.* Van Hentenryck *et al.*, 1997b).

Improvements pertain in particular to interval arithmetic-related attributes of the application. For instance, given constraints like eqn.(6) with many variables, other local consistency algorithms may perform better than the interval Newton method under particular conditions (*e.g.* bounds consistency may be faster if the constraints are monotonic). Moreover, the branching of the forward checking method could possibly be improved by using a branch-and-bound algorithm. In such case the discretization process (from interval domains to interval values) is carried out differently, *i.e.* the domains are incrementally split. This method could further be improved by using typical interval variable ordering heuristics[191] like: "round-robin" or largest domain first.

Furthermore, since constraint reasoning problems are in worst case NP-complete, the inference engine must be extended such that a (partial) solution for the setting calculation mechanism is always available. This so-called anytime behavior is necessary in many practical applications where the absence of a solution is worse than a solution that does not conform to all to all demands.

Finally, further speed ups may be realized by converging the constraint equations to other formulations like for instance Taylor expansions (*e.g.* Van Hentenryck *et al*, 1997a; Hyvönen, 1992).

9.3 Consequences for the setting calculation mechanism

The inference engine regularly generates sets of solutions. Each solution involves a value assignment for all variables present in the inference engine (*i.e.* a complete instantiation). The interface between the inference engine and the setting calculation mechanism consists of a subset of those variables, namely, those that are pres-

---

[191] In both branching methods, the domain of a variable is split until the discretization value is reached. The round-robin heuristic uses a static predetermined variable ordering cycle, that is, after the split of variable v_i the next variables v_{i+1}, v_{i+2}, ..., v_n, v_1,..., v_i, v_{i+1}, will be split. The largest domain first ordering uses a dynamic ordering based on the size of the domain, that is, the variable with the relatively largest remaining domain will be split next.

ent in both modules. Given the optimization horizon and calculation frequency of the setting calculation mechanism[192], it is assumed that the setting calculation mechanism only uses the variables that belong to the current hour instance in the time hierarchy[193].

When the setting calculation mechanism finds a sequence of devices settings that adheres to the values of this subset of variables, it complies with the complete instantiation per definition.

9.3.1 Size of the solution space

In the previous section a number of cases have been discussed that influence the size of the constraint reasoning problem and the resulting solution space that needs to be processed by the setting calculation mechanism. Given the real-time characteristics of the task of the setting calculation mechanism, it is yet unknown what its performance under conditions that involve very large solution sets (> 10,000) will be. However, to assist the setting calculation mechanism in processing large solutions sets, the solutions generated by the inference engine should be combined where possible, and ordered such that the setting calculation mechanism processes the best candidate solutions first.

As already discussed, the size of the solution space is greatly affected by the discretization values of the numerical variables. Therefore, these values should be carefully chosen since every additional value may double the solution space. For intermediate variables this is especially crucial, their discretization values should be chosen based on the interval arithmetic-related characteristics of constraint relation and their dependency on the other argument variables in the constraint relation. Determining an effective discretization value for the intermediate variables can be carried out during the tuning experiments.

---

[192] To recapitulate: in chapter six it has been discussed that: *i*) the optimization horizon of the setting calculation mechanism is assumed to be one hour, and *ii*) it generates device settings with a frequency of one per minute. Assumed that the inference engine generates only one set of solutions per hour, the setting calculation mechanism calculates sixty sequences of device settings (each conforming to at least one solution).

[193] Other variables (*i.e.* variables that are situated at more aggregated levels in the time hierarchy) may be used as fixed parameters in the models in the setting calculation mechanism. That is, they are not part of the search space of the setting calculation mechanism (if there is some uncertainty in these variables, for instance *LAI* = [2.6, 2.8], their midpoint values can be used).

9.3.2 Sets of alternative solutions

In constraint reasoning terminology, a solution implies a unique value assignment to the variables in a constraint reasoning problem (*i.e.* the inference engine). However, since the setting calculation mechanism uses only a subset of the variables processed in the inference engine, some of the solutions generated by the inference engine are equal from the viewpoint of the setting calculation mechanism. Given this notion of equivalence, the solutions generated by the inference engine can be arranged into clusters of solutions that are identical for the setting calculation mechanism. The number of clusters represents the number of so-called *'unique solutions'* for the setting calculation mechanism. Each unique solution has a certain support, *i.e.* the size of the cluster it represents.

Table 9-1 shows five solutions generated by the inference engine. Solutions 1 and 2 belong to the same set, that is, to the same the unique solution. This unique solution has a support of 2. Solution 3 belongs to another unique solution, with a support of 1. Solutions 4 and 5 also belong to the same unique solution (with a support of 2).

Table 9-1 Example solutions generated by the inference engine where hour$_t$ and hour$_{t+1}$ belong to 'morning$_{day=x}$'.

Sol.	hour$_t$			hour$_{t+1}$			morning$_{day=159}$		
no.	T	I_0	CO$_2$	T	I_0	CO$_2$	T	I_0	CO$_2$
1	[20,0 20.1]	150	[375,400]	[19.9,20.0]	[170, 180]	[375,400]	[19.7,19.8]	[170, 180]	[425,475]
2	[20,0 20.1]	150	[375,400]	[19.9,20.0]	[180, 190]	[375,400]	[19.7,19.8]	[170, 180]	[425,475]
3	[19.9,20.0]	150	[375,400]	[19.9,20.0]	[170, 180]	[375,400]	[19.7,19.8]	[170, 180]	[425,475]
4	[20,0 20.1]	150	[425,450]	[19.9,20.0]	[180, 190]	[375,400]	[19.7,19.8]	[170, 180]	[425,475]
5	[20,0 20.1]	150	[425,450]	[20.0,20.1]	[180, 190]	[375,400]	[19.7,19.8]	[170, 180]	[425,475]

Since the solutions generated by the inference engine include variables whose domains are predicted, it can only afterwards be known whether a solution was really valid. For instance, if (for some reason) the setting calculation mechanism chooses to implement solution 3 from Table 9-1 at hour$_t$ (*i.e.* it found a sequence of device settings for this solution), and afterwards it was established that the light intensity at hour$_{t+1}$ was not in [170, 180] but in [180,190], it implemented a solution that might be invalid in a constraint reasoning sense but is actually only sub-optimal with respect to the practical management problem under consideration. To prevent this as much as possible, the unique solutions can be ranked according to their 'safeness'. The safeness property of a unique solution is closely related to the support property. The safeness property refers to the number of external situations (that are different instantiations of external variables) that is covered by a unique solution. The unique solution that covers solutions 1 and 2 in Table 9-1 is safer than the unique solution that covers solutions 4 and 5, since the latter refers to one external situation while the former refers to two external situations[194].

9.3.3 Attainability of a unique solution

The unique solutions provided by the inference engine make up the solution space in which the setting calculation mechanism must find (and implement) a sequence of device settings that is consistent with one of the unique solutions[195]. Each unique solution is the result of a number of assumptions that affect the validity of these solutions w.r.t. reality. These assumptions are the following. First, each unique solution is based on the representation of the current state of the greenhouse–system in the time hierarchy. Second, each unique solution includes predictions of future (external) states. Third, each unique solution is generated by means of a set of constraints (models). These assumptions determine for a large part whether a particular unique solution can be processed successfully by the setting calculation mechanism.

When the setting calculation mechanism processes a unique solution, it carries out two tasks. First, it must find a sequence of device settings[196] that adheres to the unique solution. During this activity it uses both its own models, procedures and optimization criteria, as well as recent information gathered from the actual greenhouse–crop system. It is important that the models and (parameter) data used by the setting calculation mechanism agree with the models and data used by the inference engine. When they differ, it may not be possible to find a sequence of device settings for (some of) the unique solutions. Consequently, the inference engine "must have a clear notion of the capabilities of the setting calculation mechanism". This is especially true for the models at the hour level, because at this level setting

---

[194] Here, the external situations at the hour level and those at more aggregated levels are treated in equal way, that is, differences in external situations at the 'morning' level and differences at the 'hour' level are simply added to compute the total number. Also, since some external situations may be more likely than others probability distributions over the external situations could in principle be used.

[195] In the remainder, it is assumed that the safest unique solution will be used to generate a sequence of device settings. However, the order in which the unique solutions are processed, the number that will be processed (incl. stopping criteria) and the performance criteria used for the final selection of a sequence of device settings can be adjusted.

[196] Assuming a (rolling) planning horizon of one hour, the setting calculation mechanism computes sets of device settings with a one minute frequency. After implement of the n^th^ set of device settings, it re-computes the $(n+1)^{th}$ set using measurements taken upon implementation of the n^{th} set.

calculation mechanism and inference engine "meet". The models that relate outside climate conditions with greenhouse climate states (at the hour level) are especially vulnerable as the output of the relatively coarse models in the inference engine must correspond closely to the output generated by the more detailed models in the setting calculation mechanism.

Second, after calculation, the setting calculation mechanism must implement the first device settings out of the calculated sequence. During this implementation step the validity of all assumptions will be put to the test, because the computerized representation of the greenhouse–crop system and its real world counterpart meet during this implementation.

Both problems require substantial testing to be certain that the models used, are sufficient in the light of their respective tasks. In addition, on-line and user-based identification methods will be necessary to keep the computerized representation of the greenhouse–crop system in line with its real world counterpart. Reducing the differences between the computerized representation of the greenhouse–crop system and its real world counterpart is a major challenge in all approaches that are based on "absolute quantities" (*i.e.* like most approaches discussed in chapter four).

9.4 Grower–System interaction during daily operational management

9.4.1 Entering settings

The experiments of chapter eight simulate some of the possibilities during daily operational management of the state of the greenhouse–crop system. It has been shown that the grower has various possibilities to enter settings, that is, he can constrain crop states (and carry out a kind of direct crop control) and have the system derive the necessary environmental conditions to satisfy his settings. Furthermore, he can use qualitative, cumulative and average constraints and use the time-hierarchy to zoom in at particular levels and specify constraints there. Moreover, he can use preference distributions to state that some intervals and some qualitative values are preferred over others.

As much of this input is different from his present way of entering settings, the grower has to familiarize himself with the system. During such a learning phase he has to gain insight into attainable values for the variables that he wants to constrain, and know the external conditions during which these values occur. Furthermore, he should understand how variables are related, that is, he should - at a qualitative level - know how the variables in the models are related[197]. Knowing how the variables in the models are related helps in finding appropriate settings.

One should realize that changes in settings for numerical variables are always small, they are usually in the order of 1 to 10% of their absolute values. Growers are presently accustomed to such small changes (*e.g.* a change in a temperature setting from 20°C to 20.5°C reflects a 2.5% change[198]).

Knowing how the variables in the models are related is also helpful for realizing a smooth transition between preference levels (*i.e.* this may prevent abrupt shifts from preference level one to very low preference levels if external conditions are slightly more difficult for the crop). It is clear that determining what settings to enter into the system remains a demanding task that requires expert knowledge.

An important aspect during settings entering is the amount of uncertainty in future external conditions the grower wishes to take into account. Chapter eight showed that if there is much uncertainty 'anything goes', that is, under particular external conditions solutions can be found, whether these conditions actually occur is another matter. Finding the 'right' amount of uncertainty to take into account should partly be supported by the system[199].

Another important aspect is the behavior of the control devices. For instance, growers generally don't like to see bang-bang control behavior of the ventilation windows. When the grower instructs the supervisory control system to realize certain values, it is the task of the system to determine how to satisfy the settings of the grower. It could well be that the grower also wants to constraint the behavior of the system during the realization of these values. For instance, the change in window aperture may not be too large between successive control instances. Entering this type of behavioral constraints must be possible, however, owing to the nature of these constraints they only play a role within the setting calculation mechanism. It is clear however that by entering these kinds of restraints the grower limits the flexibility and the short term optimization potential of the setting calculation mechanism.

---

[197] For instance: 'the light intensity generally has a bigger influence on the photosynthetic production than the CO_2-concentration' or 'the transpiration rate during nighttime is approximately 10% of the daytime transpiration'.

[198] This is already the case in presently available control systems but it will also apply to 'new' variables like afternoon photosynthesis (see the examples in section 8.5).

[199] The system should for instance be able to derive from the weather predictions certain uncertainty characteristics, that is, an overcast sky will allow for a much more accurate prediction of the light intensity as compared to a partly clouded sky).

9.4.2 *Display the results of the inference process*

As can be seen from the simulations in chapter eight, the inference engine may generate many solutions for future time periods. These solutions must illustratively be presented to the grower. One possibility is a graphical representation of the solution density distribution per variable over time (shows the solution density distribution of Table 8-8). The figure exemplifies the safest solution and shows the solution distribution in shades of gray. Such a graph can be constructed for every variable that interests the grower. Moreover, the grower should be able to configure the graph, for instance, he may want to: *i*) vary the timestep of the time axis, *ii*) show multiple variables in the same graph, or *iii*) leave out the solution density distribution, etc..

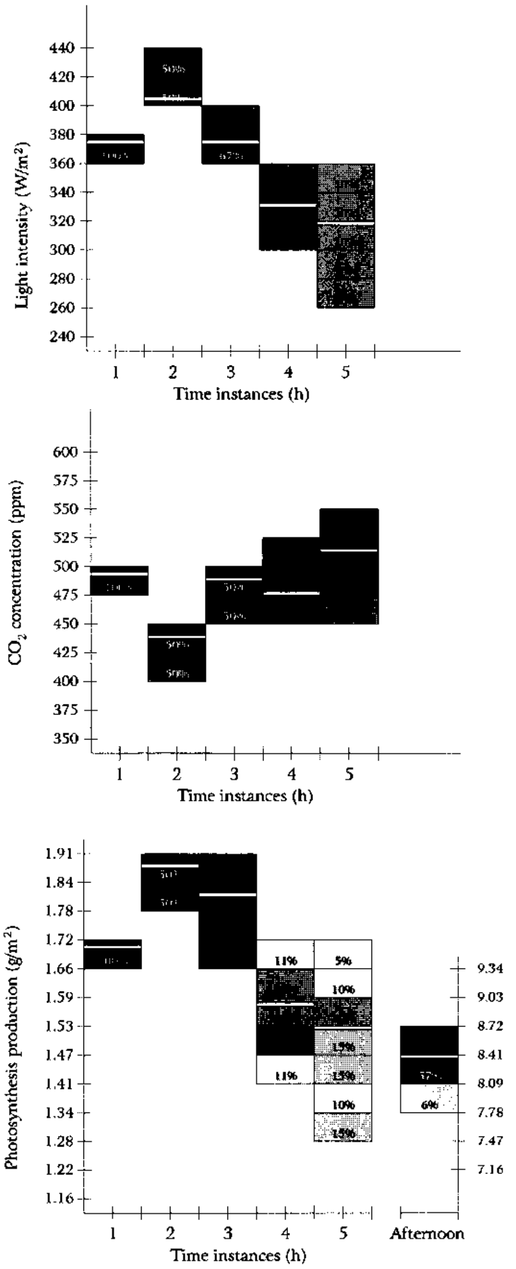


Figure 9-1 Solution density and safest solution.

Other, for instance, tabular representations are of course also needed, they can for instance be used for presenting all kinds of information about the system. Tabular representations are less useful for representing large solution sets (like the ones generated in section 8.5).

Since many of the presentation issues are of a high practical nature, they will not be discussed further. They should be addressed at a later stage in close collaboration with the growers.

9.5 Other system uses: scenario analysis

In chapter six it has been stated that the grower should also be able to use the system for scenario analysis. Scenario analysis at the operational management level offers insight in the course of the greenhouse-crop system under different weather scenario's. In this case the grower is interested in possible crop states for approximately the next week. During scenario analysis the grower is interested in the more slowly changing crop characteristics like: the risk of getting BER, the LAI, the cropload[200], etc., and not in the details within a particular day. During this activity his objective is to analyse his set of settings (and alternative sets) under various weather scenario's and inspect their outcome.

During the scenario analysis the inference engine could be used in a slightly different fashion as opposed to normal operational management. First, given the large impact of uncertainty in the external conditions on the performance of the inference engine, the scenarios will be free of this type of uncertainty (moreover, by carrying out several likely weather scenario's, it is assumed that the grower will not need full uncertainty processing). Second, since responsiveness of the system is important during scenario analysis, the inference engine will use aggregate or summary models at the higher time hierarchy levels (as discussed in section 9.2.1), thereby avoiding the hour level in the time hierarchy and thus reducing the size of the constraint reasoning problem considerably[201]. Finally, since the computations carried out during scenario processing should not interfere with normal operational management, the scenario studies should be carried out on a separate datamodel (time hierarchy) into which the actual state of the systems and the grower's settings are copied.

9.6 Concluding remarks

The above sections, together with the previous three chapters, showed an in-depth discussion of *a* next generation supervisory control system for operational greenhouse crop management. The main conclusion from this presentation may be the realization that developing a next generation supervisory control system is not a trivial undertaking. Indeed, within the architecture described above many (practical) problems remain to be solved. Fortunately, we did obtain a clearer understanding of where these problem reside, this insight facilitates working on these problems.

---

[200] Number of fruits per m^2, or per shoot.

[201] It is assumed that such is possible, that is, that the system will contain aggregate or summary models for all processes that are of importance during the scenario analysis task.

It has been shown that constraint reasoning has interesting properties in the light of the problem of operational crop production management. Also, the behavior of the prototype, as so far it has been tested, suffices in relation to the objectives set forth. It is clear that the present prototype should be seen as a first step, additional work is needed to achieve a fully operational prototype.

10. GENERAL DISCUSSION

10.1 Introduction

In this chapter we look back on the research that underlies this thesis. In chapters five and nine, the analysis, respectively the design part of this thesis have already been discussed and evaluated. Therefore we focus on: *i*) themes that link the analysis and design parts together, and *ii*) the future developments that are needed to realize an implementation in practice.

10.2 Review of the thesis objectives

The overall objective, set forth in the introduction of this thesis, was the realization of a design of a computerized support system under the constraints that such a design: *i*) is realistic w.r.t. its general application conditions, *ii*) can be applied within a broad scope of specific situations, as well as *iii*) should be reachable in the nearby future.

10.2.1 General application conditions

The application conditions of the supervisory control system refer both to the intrinsic demands of the problem solving approach applied in the proposed system, as well as to the task environment in which the system must eventually be embedded.

As most important part of the task environment, the grower's decision making style and his ways of managing crop production stand out. Since designing an *appropriate* support system has been a central theme in this thesis, it is argued that the design meets the problem characteristics discussed in chapter two, and the requirements specified in chapter five and six.

Another important issue w.r.t. the task environment is the system's anticipated operation and performance, especially under extreme weather conditions. The shift in responsibility, and change in setting type from action- and boundary-oriented settings to state-oriented settings (sections 5.3, 9.4.1 and 10.3) requires a method that guarantees robust behavior. In this thesis it is argued that the use of a preference distribution fulfills this need. Additionally, it is assumed that the necessary changes in the grower's working methods (sections 9.4.1 and 10.3) are acceptable and can be mastered.

Regarding the intrinsic demands of the problem solving approach it has been shown in chapter seven that within the requirements set forth it is indeed possible to model (the most important part of) the decision problem as a constraint reasoning problem. Because within the chosen constraint reasoning configuration:

- different types of knowledge can be accommodated,
- one can support robust behavior by using the preference mechanism as explained,
- an integrated treatment of variables at the actuator, climate and crop level through a hierarchical treatment of time is possible, and
- the output constraints that are generated, can be used within a setting calculation mechanism.

However, it remains uncertain at this early stage whether, under practical conditions, the speed of resolution of the chosen constraint reasoning approach is adequate enough (section 9.2). Extensive experimentation in practice will be needed to evaluate this aspect in the future.

10.2.2 System scope

The scope of applicability of the system concerns several aspects related respectively to the grower, the crop, the greenhouse type and the greenhouse location.

Within 'grower' aspect, the variations in decision making style and objectives of growers have been analyzed. Given the diversity in input 'modes', that is, the variety of decision variables (as discussed in various places in part two of this thesis), it is argued that the approach offers ample opportunities for the different styles of decision making found among growers.

As to the 'crops' aspect it is argued that the approach is most useful for crops that display a certain difficulty w.r.t. the realization of the operational objectives of the grower. These crops exhibit 'challenges' like: realizing a proper balancing of the crop's vegetative and generative growth, maintaining a good product quality, etc.. These challenges should fall within the scope of operational management (*i.e.* they can be influenced within a time frame of three to five days). Target crops typically have a moderately long to very long growing season, possibly stretching out over more than one year. A long growing season can give rise to a cumulative effect of errors made in operational management.

A serious constraint on the applicability of the approach for a particular crop is the availability of an adequate body of knowledge about the crop. That is, important crop-specific processes and problems like the types discussed above, should be sufficiently known, so that this knowledge can be built into the system. Given the available literature on the vegetable crops tomato, sweet pepper and cucumber, it is assumed that in these cases the amount and quality of knowledge is satisfactory for its incorporation in the knowledge base of the proposed supervisory control system.

The aspects 'greenhouse type' and 'geographical location' are considered together because the applicability and value of the approach depends upon the controllability of the environment of the crop. In situations or in greenhouses where precise control of the environment of the crop is not possible, the value of the approach is greatly reduced. This for instance applies to production conditions in the Mediterranean area (where sufficient cooling capacity is usually not available). In these cases the purpose of the system is reduced to providing information, which is a valuable asset on its own.

10.2.3 Introduction in practice

An important objective was a possible initial implementation of the proposed system in the nearby future. If we limit 'the nearby future' to, for instance, five years, it may be possible to comply with this requirement. That is, it is argued that there are no *fundamental* limitations that impede the implementation of the supervisory control system. However, the amount of work to realize this goal is impressive. Consider for instance the need for a modular knowledge base in which tomato-specific knowledge can easily be replaced with knowledge of other crops that resemble the production characteristics of a tomato crop. Design and implementation will occupy a highly skilled knowledge engineer for at least a year.

In this light complete realization of the overall objective may be seen as difficult to attain, especially since re-use of the control logic (*i.e.* the software) of presently available control systems is difficult (given the fundamental differences between the supervisory control system and the presently available control systems).

Fortunately, within the design and implementation of the new system, re-use of third-party components remains a viable means to reduce implementation time (if one is able to take the 'not invented here' hurdle). Significant speedups can be realized by leveraging off mainstream technology like: user interface components, constraint solvers, database management systems, basic data acquisition and control software, etc..

10.3 Differences with currently available systems

An important difference between the grower's present control systems and the supervisory control system proposed here, is the shift in responsibility. In the new system the grower essentially instructs the system *what* it should achieve and not (or only partly) *how* it should behave. This change in approach reduces the perceived predictability of the system's behavior, which is, in present systems not only a means to judge their operation and performance but also an implicit check on equipment breakdown.

Transferring (part of) the responsibility of setting determination to the proposed system requires the grower to trust its behavior. Gaining the grower's trust is of course not easy and requires:

- the system to display acceptable behavior in the eyes of the grower,
- the possibility to learn using this new system, (gradually from his current practices to new ways of working and through examples), and
- sufficient support from experts (*e.g.* consultants from extension service agencies) during start up. An extensive coaching program for the introduction of the new system will be necessary.

A second significant difference between currently available systems and the approach proposed in this work, is the responsibility of the grower for keeping the computerized representation of the greenhouse–crop system in agreement with its real world counterpart. This is a new and important activity since it directly affects the behavior of the system and, as a consequence, the attitude of the grower towards it. Although in professional literature a certain indifference w.r.t. measuring and maintenance of measuring equipment has been reported, it is believed that with proper training and motivation (*e.g.* through warning messages from the system) difficulties can be overcome in a manner that it will not hamper the realization of the approach.

10.4 Experiences should reflect on future developments

From the background of a systems approach in corporate decision making In 't Veld (in Hofstede, 1992) argues that any solution to a problem which is not totally unfeasible will work (or can be made to work). He also argues that given the lack of a reference or control situation, the actual quality of a solution can never be fully assessed.

In chapter two it has been shown that the key notion with respect to feasibility of a solution proposal is its "context". That is, any computerized support system that is designed without sufficient regard of application conditions will fail. In this light it can be explained why proposals in section 4.4 have not yet resulted into successful implementations in practice. The reason is that they can be characterized as prescriptive approaches that deal with decisions that require the judgment of the grower. By doing so they overstep a bound that has time and time again demonstrated its importance.

Unfortunately, in management and decision research, prescriptive approaches have been much more attractive than descriptive ones, mainly because they are simpler. Among others they lack the arbitrariness and opportunism of real world problem situations. They also reward the researcher with a sense of value, because the advocated method performs better than its control in the light of the (often impractical or sometimes even unrealistic) assumptions and preconditions. However, apart from their role in scientific discovery (*e.g.* in sensitivity studies) or in carefully specified circumstances, prescriptive approaches - as a general guideline - should be used with utmost care. This is why in this work a descriptive approach precedes a prescriptive one. In a descriptive approach both the insight in the problem structure and the task environment, as well as awareness of the available means are fundamental. On these analyses a prescriptive approach can then be constructed.

If one distinguishes between structured and programmable aspects, this thesis is essentially a contribution towards:

- the clarification of the problem structure, and
- the normalization of a part of the resolution process (*i.e.* making parts programmable) by providing a computational approach consistent with the perceived structure.

The effort of normalization does not imply that one must end up with a programmed approach in which all possible actions in all possible states are evaluated beforehand. Indeed, the normalization of the decision process (its readiness to be programmed) is still poorly developed, since it is not feasible to identify and formulate indisputably and exhaustively: all objectives, criteria, variables, constraints and processes constituting the decision problem.

The normalization of any decision process must take into account the subjective aspects, this is why - in the present case - the involvement of the grower is so important. It is therefore argued, that owing to the nature of the problem domain (*i.e.* the complexity of the tasks of the grower and the knowledge available in the domain of crop production), a fully automated resolution process will never realize the same level of efficacy that can be realized with an approach in which the grower is purposely involved.

10.5 Further work

In this section the need for further work is discussed. On one hand it addresses the need for more detailed information on a number of aspects within the scope of this thesis, and on the other hand some suggestions are made to extend and elaborate upon the ideas set forth in the quest to realize a successful implementation in practice.

10.5.1 Elaboration and validation of the analysis

The analysis of the problem of operational crop production management, as discussed in chapter two, can possibly be seen as a first step in a more elaborate study on the decision making behavior of greenhouse growers. The results of such a study could improve the model described in section 2.5. Moreover, such a study might reveal the more fundamental grounds for differences in behavior among growers (*e.g.* the differences in the frequency with which they change their settings).

Furthermore, the analysis of the knowledge available for crop production should be extended. Bringing the available knowledge together could involve both the collection of already structured knowledge as well as additional model development The latter has already been started for the problem of Blossom-end rot (appendix 1).

10.5.2 Validation of the design and extension of the framework

The design of the supervisory control system consists of both a description of an overall framework, as well as a detailed discussion on, and an implementation of the inference engine.

Validation and extension should be carried out at various levels of detail and intensity. At the level of the overall framework there is a need to:

- specify and elaborate on domain and task structure of the individual modules including the interfaces that exist between the modules, and
- validate the practicability of the overall approach with growers. Active involvement of the prospective users is a ground rule in modern software engineering practices, and can, given the current state of affairs, start immediately.

At the level of the problem solving core within the overall framework, the current prototype amongst others needs:

- additional work on the integration between the inference engine and the setting calculation mechanism. Furthermore, issues like: the use of different models describing the same processes, the size of the solution space and the processing of alternative solutions in relation to the amount of uncertainty allowed for (section 9.3), require further study.
- extension of the algorithms in order to test the inference engine on more demanding examples. In the light of the availability of commercial optimization environments like ILOG OPL Studio® (Van Hentenryck, 1998), it is advised to seriously consider using such a tool.
- additional development of methods that address the problems related to the processing and display of uncertainty while interacting with the grower (section 9.4).

10.5.3 Other issues

Projects can be characterized as *applied research* when methods, techniques or knowledge generated by the scientific community at large, are put to use in a specific context. In this section some issues are addressed that are particularly relevant to applied research projects like the one reported here and the follow-up that may result from this work.

10.5.3.1 The (re-)use of tools

Re-use is the key to efficient development. It implies: *i*) avoiding the use of third generation programming languages (*e.g.* FORTRAN) where possible, *ii*) leveraging off commercial third-party products, and *iii*) making a point of using applicable commercial software products even if this requires (minor) compromises to the vision or prospective system design.

A mass of literature and other evidence has been produced by the IT community and associated parties, that prove the point of the gains that can be achieved in this way.

Unfortunately, adequate application of this philosophy requires a fundamental change in attitude as well among designers as among users. There still is a long way to go before the advocated methods will really become general practice.

10.5.3.2 Experimentation in practice

Testing of new approaches for computerized support for operational crop production is still in its infancy. It has been shown in chapter four that the amount of evaluation for the discussed approaches has been very limited. For proper testing of a computerized support system at least the following is needed:

- the duration of the experiment or field trial should be (at least) one growing season. Short term experiments (*e.g.* Harazono, 1991) do not capture the full difficulty a grower experiences since long-term influences are ignored, and,
- results should be compared with the results obtained by a growers that use conventional methods (*i.e.* standard control systems).

10.5.4 Working towards introduction in practice

Sections 10.5.1 and 10.5.2 describe the most important issues that require further elaboration. In this section suggestions will be given on how to go about on these issues.

Schiefer (1991) stated that new computerized support systems require marketing to become a success. He also argues that the marketing approach is difficult to implement and requires a different type of people to implement it. It is no longer the developer/researcher whose main focus is the system to be designed, but a developer who looks at the prospective user. The key is to fit the needs of the user, the problem and the (institutional) environment. In this light a shift in project organization should occur when the focus of the project changes.

Given the impressive amount of work that lies ahead, a 'programme' could be set up in which the necessary projects can be embedded. A programme structure has the advantage that it ensures that the individual projects tie-up, and allows interest groups to be properly involved and embedded.

The following communities could be characterized as interest groups: *i*) growers (supported by their statutory organizations), *ii*) control system manufacturers, *iii*) research groups that carry out applied and fundamental research in relevant fields of study, *iv*) extension service agencies (including other service agencies like auctions), and *v*) government and grant organizations (that is, the ones that have a specific interest in greenhouse production *e.g.* reduction of energy consumption). Each of these interest groups should play a role in the committee that determines the course of the programme. Without underestimating the role other interest groups, it is clear that given the nature of such programme, growers and control system manufacturers are the most important interest groups.

Apart from their role in the programme committee is, each interest group should play an active role in one or more projects of the programme. Depending on their expertise, their public/organizational role and/or their interest in the expected benefits, each interest group could provide various services to the individual projects. Amongst others they may: *i)* provide funding, *ii)* carry out research, *iii)* provide data, *iv)* manage projects, *v)* make available specific knowledge and expertise, *vi)* provide test and research environments, *vii)* carry out measurements (*e.g.* for model calibration), *viii)* supply equipment, etc..

Regarding the issues referred to in sections 10.5.1 and 10.5.2, individual interest groups could play the following roles and services.

In an investigation of the operational decision making behavior of greenhouse growers, research groups may take the lead. The objective of such project is a further refinement and validation of the model described in section 2.5. As a matter of course, growers and extension service specialists should be heavily involved. Likewise, experts from control system manufacturers could assist when the use of control systems by the grower is being investigated. Given the scientific nature of such study, funding (in terms of making the necessary manpower and specialists available) should be provided by the involved parties themselves.

Further collection, elicitation and modelling of knowledge in the field of operational crop production management is a substantial undertaking (it could in fact be seen as a sub-programme in itself). However, without clear focus, tangible objectives and a good project structure and coordination there is a high risk that the individual knowledge 'blocks' do not fit together or have slight incompatibilities (*.e.g.* in the definition of concepts, representation, etc.). Furthermore, given the high cost of experimentation, efficiency in acquiring knowledge can be regarded as crucial for success. In this light almost all interest groups can contribute. For instance, already planned for experiments in research facilities and measurements in commercial greenhouses can be used to calibrate already existing models or validate new ones. Control system manufacturers can contribute by slightly adjusting current systems to make measured environmental data more easily available. Furthermore, growers can carry out additional measurements on their crop (or have them carried out by students) and provide this data, etc..

A clear focus of what knowledge must be brought together, and in what representational format it must be captured can only be established after the framework for the supervisory control system has been elaborated upon. The framework of the supervisory control system can therefore be seen as the backbone of the programme. Simultaneous with the investigation of the decision making behavior of greenhouse growers, the system's framework can be elaborated upon. Working on the architecture of the supervisory control system should be a joint effort of the control system manufacturers and a research group specialized in system design. At the same time the philosophy of the approach can be introduced to the growers by extension service specialists and control system manufacturers. On the basis of feedback from the growers adaptations of the approach may be required also, interaction with growers will help prioritize knowledge acquisition.

The implementation phase of the various modules can start when the architecture of the system has been worked out in sufficient detail. Since software development is still expensive, additional funding will be required, here grant organizations (especially the ones that strive for a reduction of energy consumption) can play a role as the overall approach is directed toward more effective management of vegetable production.

Finally, a few remarks about the chance of success of such a programme may be appropriate.

Approaches like this require sufficient start-up support, that is, an initial project team that is able to bring together the various interest groups and can persuade them (*i.e.* especially competing parties) to work together. An important motivation of commercial interest groups is their expected benefit. In this light it is essential that the architecture of the supervisory control system is sufficiently open such that commercial parties can profit from it in a way that suits their market role.

With respect to market opportunities, positive as well as negative developments could be observed. A positive consequence of the constant pressure on the market for control systems is the fact that it may stimulate cooperation in a programme as discussed above. Furthermore, the software and hardware industry is becoming more and more mature, this results in the fact that software systems are becoming commodities. They become more easy to construct using off the shelf technology. The added value lies in how one can best make use of available knowledge. The approach discussed here is in line with this development.

REFERENCES

REFERENCES

Aarts, E. and J.K. Lenstra, 1997. *Local search in combinatorial optimization*. John Wiley and Sons, Chichester, UK, 512pp.

Adams, P. and L.C. Ho, 1989. Effects of constant and fluctuating salinity on the yield, quantity and calcium status of tomatoes. *Journal of horticultural science*, vol.64, p.725-732.

Adams, P. and L.C. Ho, 1992. The susceptibility of modern tomato cultivars to blossom-end rot in relation to salinity. *Journal of horticultural science* vol.67, p.827-839.

Adams, P. and R. Holder, 1992. Effects of humidity, Ca and salinity on the accumulation of dry matter and Ca by the leaves and fruit of tomato. *Journal of horticultural science* vol.67, p.137-142.

Aikman, D.P. and G. Houter, 1990. Influence of radiation and humidity on transpiration: implications for calcium levels in tomato leaves. *Journal of horticultural science*, vol.65, p.245-253.

Allen, J.F.,1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, vol.26, nr.11, p.832-843.

Alliot, J. M. and T. Schiex, 1994. *Intelligence Artificielle et Informatique Théorique*. 2nd édition. Cepadues, France. (in French)

Anonymous, 1995. *Econaut: energiezuinig stoken zonder produktieverlies*. Hoogendoorn automatisering, 's Gravenzande, The Netherlands, 4 pp. (in Dutch)

Anonymous, 1997. *Proceedings of the annual ILOG optimization suite users' meetings*. http://www.ilog.com/html/products/optimization/customer_papers.htm

Bacchus, F. and A. Grove, 1995. Graphical Models for Preference and Utility. In:Besnard, P. and S. Hanks (eds.), *Proceedings of the eleventh conference on uncertainty in artificial intelligence. August 18-20, Montreal, Canada.*

Backus J.W.M.; A.J.J. Van Roestel and L.J.L.D.Van Griensven, 1991. Development of an expert system for the cultivation of mushrooms and validation of the prototype. *Proceedings of the 13th international congress on the science and cultivation of edible fungi. Mushroom Science XIII, vol.2.*, p.791-798.

Bailey B.J., 1990. Greenhouse models and climate control. *Proceedings of the international seminar and British-Israel workshop on greenhouse technology*, Bet Dagan, Israel, 26 March - 2 April 1990. p.79-90.

Bailey, B.J.; B.G.D. Haggett; A. Hunter; W.J. Albery and L.R. Svanberg, 1988. Monitoring nutrient film solutions using ion-selective electrodes. *Journal of agricultural engineering research*, vol.40, nr.2, p.129-142.

Bailey, B.J. and I. Seginer, 1989. Optimum control of greenhouse heating. *Acta Horticulturae*, vol.245, p.512-518.

Bakker, J.C., 1991. *Analysis of humidity effects on growth and production of glasshouse fruit vegetables*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 155 pp.

Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak, 1995. *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, 279 pp.

Bakker, J.C. and G.P.A.Van Holstein, 1995. Screens. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.185-195.

Belda, R.M. and L.C. Ho, 1993. Salinity effects on the network of vascular bundles during tomato fruit development. *Journal of horticultural science*, vol.68, p.557-564.

Bertin, N. and E. Heuvelink, 1993. Dry matter production in a tomato crop: comparison of two simulation models. *Journal of horticultural science*, vol.68, p.995-1011.

Boehlje, M.D. and V.R. Eidman, 1984. *Farm management.* Wiley, New York, USA, 806 pp.

Bontsema, J., 1995. Control principles. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.224-231.

Bot, G.P.A., 1983. *Greenhouse climate: from physical processes to a dynamic model.* PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 240 pp.

Bot, G.P.A., 1995. Greenhouse climate control: conclusions. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.241-242.

Bot G.P.A. and H. Challa, 1991. Climate control of greenhouses and crop growth. In: Bajaj, Y.P.S. (ed.) *High-tech and micro-propagation.* Springer-Verlag, Berlin, Germany, vol.1, p.516-539;

Bots, P.W.G., 1989. *An environment to support problem solving.* PhD thesis Technical University Delft, Delft , The Netherlands, 249 pp.

Bradfield, E.G. and C.G. Guttridge, 1984. Effects of night-time humidity and nutrient solution concentration on the Ca-content of tomato fruit. *Scientia horticulturae*, vol.22, p.207-217.

Breuer, J.J.G. and P. Knies, 1995. Ventilation and cooling. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.179-185.

CAMASE, 1996. *Concerted Action for the development and testing of quantitative Methods for research on Agricultural Systems and the Environment.* http://www.bib.wau.nl/camase/

Chalabi, Z.S., 1991. A generalized optimization strategy for dynamic CO2 enrichment in greenhouses. *European Journal of Operational Research*, vol.59, p.308-312.

Challa, H., 1980. A stategy for climate control in greenhouses in early winter production. *Acta Horticulturae*, vol.106, p.159-164.

Challa H.; G.P.A. Bot; E.M. Nederhoff and N.J. Van der Braak, 1988.Greenhouse climate control in the nineties. *Acta-Horticulturae*, vol.230, p.459-470.

Challa, H. and E. Heuvelink, 1993. Economic evaluation of crop photosynthesis. *Acta Horticuturae*, vol.328, p.219-228.

Challa, H. and A.H.C.M. Schapendonk, 1986. Dynamic optimization of CO_2 concentration in relation to climate control in greenhouses. In: Enoch, H.Z. and B.A. Kimball *Carbon dioxode enrichment of greenhouse crops*, CRC Press Inc., Boca raton, FL, USA., vol.1, p.147-160.

Challa, H. and G. van Straten, 1993. Optimal diurnal climate control in greenhouses as related to greenhouse management and crop requirements. In: Hashimoto Y., G.P.A. Bot; W. Day; H.J. Tantau and H. Nonami (eds.) *The Computerized Greenhouse: Automatic Control Application in Plant Production.* Academic Press, New York, USA, p.119-137.

Chamand, A.; A. Fischler; D. B. Guinaudeau and A. Guillaud, 1995. *CHIC lessons on CLP methodology*. Technical report Dassault Aviation, Saint-Cloud, France, http://www.ecrc.de/eclipse/html/CHIC_Methodology.html.

Cleary, J.G., 1987. Logical Arithmetic. *Future computing systems*, vol.2, nr.2, p.125-149.

Cockshull, K.E. and L.C. Ho, 1995. Regulation of tomato fruit size by plant density and truss thinning. *Journal of horticultural science*, vol.70, nr.3, p.395-407.

Davis, E., 1987. Constraint propagation with interval labels. *Artificial intelligence*, vol.32, nr.3 p.281-331.

Davis, G.B. and M.H. Olson, 1985. *Management information systems. Conceptual foundations, structure and development*. Second edition, McGraw-Hill, New York, USA, 693 pp.

De Jong, T., 1990. *Natural ventilation of large multi-span greenhouses*. Ph.D. thesis Wageningen Agricultural University, Wageningen, The Netherlands, 116 pp.

De Kleer, J. and J.S. Brown, 1984. A qualitative physics based on confluences. *Artificial Intelligence*, vol.24, p.7-83.

De Koning, A.N.M., 1988. The effect of different day/night temperature regimes on growth, development and yield of glasshouse tomatoes. *Journal of horticultural science*, vol.63, nr.3, p.465-471.

De Koning, A.N.M., 1989. Development and growth of a commercially grown tomato crop. *Acta Horticulturae*, vol.260, p.267-273.

De Koning, A.N.M., 1994. *Development and dry matter distribution in glasshouse tomato: a quantitative approach*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 240 pp.

De Koning, A.N.M. and J.C. Bakker, 1991. In situ plant weight measurement of tomato with an electronic force gauge. *Acta Horticulturae*, vol.304, p.183-186.

Dechter, R. and J. Pearl, 1989. Tree Clustering for Constraint Networks. *Artificial Intelligence*, vol.38, nr.3, p.353-366.

Dinkel, J.J.; M.J. Tretter and D. Wong, 1988. Interval Newton methods and perturbed systems. *Applied mathematics and computation*, vol.28, p.211-222.

Dubois, D. and H. Prade, 1986. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, USA.

Ehler, N. and P. Karlsen, 1993. OPTICO: a model based real-time expert system for dynamic optimization of CO2 enrichment of greenhouse vegetable crops. *Journal of horticultural science*, vol.68, nr.4, p.485-494.

Ehret, D.L. and L.C. Ho, 1986a. Translocation of Ca in relation to tomato fruit growth. *Annals of botany*, vol.58, p.679-688.

Ehret, D.L. and L.C. Ho, 1986b. The effects of salinity on dry matter partitioning and fruit growth in tomatoes grown in nutrient film culture. *Journal of horticultural science*, vol.61, p.361-367.

Ehret, D.L. and L.C. Ho, 1986c. Effects of osmotic potential in nutrient solution on diurnal growth of tomato fruit. *Journal of experimental botany*, vol.37, p.1294-1302.

Elzas, M.S., 1984. System paradigms as reality mappings. In: Ören, T.I.; B.P. Zeigler and M.S. Elzas, *Simulation and model-based methodologies: an integrative view*. Springer-Verlag, Berlin, Germany, p.41-67.

Elzas, M.S., 1986. The applicability of artificial intelligence techniques to knowledge representation and simulation. In: Elzas, M.S.; T.I. Oren and B.P. Zeigler, *Modelling* *and simulation in the artificial intelligence era*. North Holland, Amsterdam, The Netherlands, p.19-40.

Fargier, H.; J. Lang; R. Martin-Clouaire and J.P. Rellier, 1994. Uncertainty and flexibility in constraint satisfaction: a case study and an application to agricultural planning. *Proceedings of ECAI'94, workshop on constraint satisfaction issues raised by practical applications*. Morgan-Kaufmann, Amsterdam, The Netherlands.

Forbus, K.D., 1984. Qualitative process theory. *Artificial Intelligence*, vol.24, nr.1-3, p.85-168.

Freuder, E.C., 1982. A sufficient condition of backtrack-free search. *Communications of the ACM*, vol.29, nr.1, p24-32.

Freuder, E.C. and R.J. Wallace, 1992. Partial constraint satisfaction. *Artificial Intelligence*, vol.58, nr.1-3, p.21-70.

Fynn, R.P.; W.L. Roller and H.M. Keener, 1989. A decision model for nutrition management in controlled environment agriculture. *Agricultural systems*, vol.31, nr.1, p.35-53.

Gal, S.; A. Angel and I. Seginer, 1984. Optimal control of greenhouse climate: methodology. *European journal of operational research*, vol.17, p.45-56.

Gamma, E.; R. Helm; R. Johnson and J. Vlissides; 1994. *Design patterns, elements of reusable object-oriented software*. Addison Wesley, Workingham, UK, 395 pp.

Garey, M.R. and D.S. Johnson, 1979. *Computers and intractability, guide to the theory of NP-completeness*. W.H. Freeman and company, San Francisco, USA, 338 pp.

Gaschnig, J., 1977. A general backtrack algorithm that eliminates most redundant checks. *Proceedings IJCAI-77*. Cambridge, MA, USA, p.457-457.

Gauthier, L., 1992. GX: a Smalltalk-based platform for greenhouse environment control. Part 1: Modelling and managing the physical system. Part 2: Supporting and implementing control strategies. *Transactions of the ASAE*, vol.35 nr.6, p.2003-2020.

Gauthier, L., 1993. Deployment of a knowledge-based, object-oriented real-time climate control system in greenhouses. *Proceedings of IJCAI Workshop on Artificial Intelligence in Agriculture, Natural Resources and Environmental Sciences*. IJCAI, Chambery, France, p.11-18.

Gent, I.P. and T. Walsh, 1996. The satisfiability constrain gap. *Artificial Intelligence*, vol.81, nr.1-2.

Giarratano, J.C. and G. Riley, 1989. *Expert systems: principles and programming*. PWS-Kent publishing company, Boston, USA, 632 pp.

Gieling, T.H. and K. Schurer, 1995. Sensors and measurements. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, p.211-222.

Gieling T.H.; E. Van Os and A. De Jager, 1988. The application of chemo-sensors and bio-sensors for soil-less cultures. *Acta Horticulturae*, vol.230, p.357-361.

Goudriaan, J., 1977. *Crop micro-meteorology : a simulation study*. PhD thesis Wageningen Agricultural University (LH-683), Wageningen, The Netherlands, 249 pp.

Graham, I.; 1994. *Object oriented methods*. Second Edition. Addison Wesley, Workingham, UK, 493 pp.

Guay, R. and L. Gauthier, 1991. Knowledge representation in a tomato disorder diagnosis system. *Computers and electronics in agriculture*, vol.6, nr.1, p.21-32.

Guerrin, F., 1991. Qualitative reasoning about an ecological process: interpretation and in hydro-ecology. *Ecological modelling*, vol.59, nr.3-4, p.165-201.

Gutman, P.O.; P.O. Lindberg; I. Ioslovich and I. Seginer, 1993. A non-linear optimal greenhouse control problem solved by linear programming. *Journal of agricultural engineering research*, vol.55, p.335-351.

Hamscher, W.; L. Console and J. De Kleer, 1992. *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., San Mateo, USA, 570 pp.

Hansen, E.R., 1992. *Global optimization using interval analysis*. Marcel Dekker Inc., New York, USA, 230 pp.

Hansen, E.R. and R.I. Greenberg, 1983. An interval Newton method. *Applied mathematics and computation*, vol.12, p.89-98.

Hanson, R.J., 1968. *Interval arithmetic as a closed arithmetic system on a computer*. Jet Propulsion Laboratory report 197.

Haralick, R. and G. Elliott, 1980. Increasing tree search efficiency for constraint satisfaction. *Artificial Intelligence*, vol.14, p.263-313.

Harazono Y., 1991. Artificial intelligence technology to control plant environments. In: Bajaj, Y.P.S. (ed.) *High-tech and micro-propagation*. Springer-Verlag, Berlin, Germany, vol.1, p.432-451.

Harazono Y.; H. Kamiya and K. Yabuki, 1988. A control method based on an artificial intelligence technique and its application for controlling plant environment. *Acta Horticulturae*, vol.230, p.209-214.

Hartog, R.J.M. and A.J.M. Beulens, 1993. Constraint network specification for interactive planning. *Proceedings of the first European congress on fuzzy and intelligent technologies (EUFIT'93)*. Aachen, Germany, 7-10 September 1993 , vol.2, p.831-838.

Hashimoto Y., 1980. Computer control of short term plant growth by monitoring leaf temperature. *Acta Horticulturae*, vol.106, p.139-146.

Hashimoto Y., 1991. Computer integrated plant growth factory for agriculture and horticulture. *Proceedings of the 1st IFAC/ISHS workshop on mathematical and control applications in agriculture and horticulture*. Pergamon press, Oxford, UK. p.105-110.

Hashimoto, Y.; T. Morimoto and S. Funada, 1981. Computer processing of speaking plant for climate control and computer aided plantation (computer aided cultivation). *Acta Horticulturae*, vol.115, p.317-325.

Hashimoto, Y.; T. Morimoto and T. Fukuyama, 1985. The speaking plant approach to the synthesis of control system in the greenhouse. *Acta Horticulturae*, vol.174, p.219-226.

Hayes, B., 1997. Can't get no satisfaction. *American Scientist*, vol.85, nr.2, p.12-19.

Hayes-Roth, F.; D.A. Waterman and D.B. Lenat, 1983. *Building expert systems*. Addison Wesley, Reading, UK, 444 pp.

Heijnen, J.G.; T. Buitelaar and H. De Kroon, 1982. *Kasklimaatregeling*. Second edition. CST, Ede, The Netherlands 322 pp. (in Dutch)

Heuvelink, E., 1995. Dry matter partitioning in a tomato plant: one common assimilate pool? *Journal of experimental botany*, vol.46, p.1025-1033.

Heuvelink, E., 1996. *Tomato growth and yield: quantitative analysis and synthesis*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 326 pp.

Hilhorst, R.A., 1992. *Kennissystemen in de agrarische sector*. QR Agrimathica, Den Haag, The Netherlands, 44 pp. (in Dutch)

Ho, L.C. and P. Adams, 1994. The physiological basis for high fruit yield and susceptibility to calcium deficiency in tomato and cucumber. *Journal of horticultural science*, vol.69, p.367-376.

Ho, L.C.; P. Adams; X.Z. Li; H. Shen; J. Andrews and Z.H. Xu, 1995. Responses of Ca-efficient and Ca-inefficient tomato cultivars to salinity in plant growth, calcium accumulation and Blossom-end rot. *Journal of horticultural science*, vol.70, p.909-918.

Ho, L.C.; R.M. Belda; M. Brown; J. Andrews and P. Adams, 1993. Uptake and transport of calcium and the possible causes of Blossom-end rot in tomato. *Journal of experimental botany*, vol.44, p.509-518.

Ho, L.C.; R.I. Grange and A.J. Picken, 1987. An analysis of the accumulation of water and dry matter in tomato fruit. *Plant, cell and environment*, vol.10, nr.2, p.157-162.

Hofstede, G.J., 1992. *Modesty in modelling: on the applicability of interactive planning systems with a case study in pot plant cultivation*. Thesis Publishers, Amsterdam, The Netherlands, 243 pp.

Hoshi, T. and T. Kozai, 1984. Knowledge-based and hierarchically distributed online control system for greenhouse management. *Acta Horticulturae*, vol.148, p.301-308.

Huijs, J.P.G., 1995. Supplementary lighting. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.202-205.

Huirne, R.B.M., 1990. *Computerized management support for swine breeding farms*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 165 pp.

Hyvönen, E., 1992. Constraint reasoning based on interval arithmetic: the tolerance propagation approach. *Artificial Intelligence,* vol.58, nr.1-3, p.71-112.

Jacobson, B.K.; Jones, J.W. and P. Jones, 1987. *Tomato greenhouse environment controller: real-time expert system supervisor*. ASAE summer meeting paper no 87-5022.

Jolliet, O. and B.J. Bailey, 1992. The effect of climate on tomato transpiration in greenhouses: measurements and models comparison. *Agricultural and forest meteorology*, vol.58, nr.1-2, p.43-62.

Jones, P.; Jacobson, B.K. and J.W. Jones, 1988. Applying expert systems to real-time greenhouse controls. *Acta Horticulturae*, vol.230, p.201-208.

Keen, P.G.W. and M.S. Scott Morton, 1978. *Decision support systems, an organizational perspective*. Addison Wesley, Reading, UK, 264 pp.

Keene, S.; 1989. *Object oriented programming in Common Lisp, a programmer's guide to CLOS*. Addison Wesley, Workingham, UK, 266 pp.

Kerssies, A., 1994. *Epidemiology of Botrytis spotting on Gerbera and Rose flowers grown under glass*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 133 pp.

Kettenis, D.L., 1990. *Simulatie met continu en discrete modellen*. Stenfert Kroese, Leiden, The Netherlands, 188 p. (in Dutch)

Klir, G.J. and T.A. Folger, 1988. *Fuzzy sets, uncertainty, and information*. Prentice Hall, London, UK, 355 pp.

Kokeny, T.; C. Le Pape; E. Jacquet-Lagrèze; J. Chang; M. Shuttleworth; N. Kosmatopoulos; G. Kostaras; J.M. David; P. Wolff; *CHIC-2 (ESPRIT Project P-22165) Methodology Survey*. 61 pp.

Kozai, T., 1985. Ideas of greenhouse climate control based on knowledge engineering techniques. *Acta Horticulturae*, vol.174, p.365-373.

Kozai, T. and T. Hoshi, 1989. Intelligent information systems for production management in agriculture and horticulture. *Future generation computer systems*, vol.5, p.131-136.

Krause, P., and D. Clark, 1993. *Representing uncertain knowledge : an artificial intelligence approach*. Kluwer, Dordrecht, The Netherlands, 277 pp.

Kumar, V., 1992. Algorithms for constraint satisfaction problems: a survey. *AI magazine*, vol.13, nr.1, p.32-44.

Lange, D. and H.J. Tantau, 1996. Climate management for disease control investigations on control strategies, plant densities and irrigation systems. *Acta Horticulturae*, vol.406, p.105-113.

Latour, B., 1988. *Science in action*. Harvard university press, Boston, USA, 274 pp.

Lees, M.J.; J. Taylor; A. Chotai; P.C. Young and Z.S. Chalabi, 1996. Design and implementation of a proportional-integral-plus (PIP) control system for temperature, humidity and carbon dioxide in a glasshouse. *Acta Horticulturae*, nr.406, p.115-123.

Leutscher, K.J., 1995. *Operational management in pot plant production*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 285 pp.

Mackworth, A.K., 1987. Constraint satisfaction. In: Shapiro, S.C. (ed), *Encyclopaedia of artificial intelligence, vol.1*. Wiley Interscience Publishers, NY, USA.

Marcelis, L.F.M., 1994. *Fruit growth and dry matter partitioning in cucumber*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 173 pp.

Martin-Clouaire, R., 1993. CSP techniques with fuzzy linear constraints: practical issues. *Proceedings of the first European congress on fuzzy and intelligent technologies (EUFIT'93)*. Aachen, Germany, 7-10 September 1993, p.1117-1123.

Martin-Clouaire R.; Boulard T.; Cros M.-J. and B. Jeannequin, 1993a. Using empirical knowledge for the determination of climatic setpoints: an artificial intelligence approach. In: Hashimoto Y., G.P.A. Bot; W. Day; H.J. Tantau and H. Nonami (eds.) *The Computerized Greenhouse: Automatic Control Application in Plant Production*. Academic Press, New York, USA, p.197-224.

Martin-Clouaire R.; Kovats K. and M.J. Cros, 1993b. Determination of greenhouse climate setpoints by SERRISTE: the approach and its object-oriented implementation. *Artificial intelligence applications in natural resources, agriculture and environmental science*, vol.7, nr.1, p.1-15.

Martin-Clouaire, R.; P.J. Schotman and M. Tchamitchian, 1996. A survey of computer-based approaches for greenhouse climate management. *Acta Horticulturae*, vol.406, p.409-423.

McKinion, J.M.; D.N. Baker; F.D. Whisler and J.R. Lambert,1989. Application of the GOSSYM/COMAX system to cotton crop management. *Agricultural systems*, vol.31, nr.1, p.55-65.

Meyer, B., 1995. On the role of methodology. In: Alagar, V.S. and R. Missaoni (eds.). *Object-oriented technology for databases and software systems*. World Scientific, Singapore. p.1-6.

Minsky, M., 1965. *Matter, Mind and Models*. Technical Memo AIM-77, MIT, Artificial Intelligence Laboratory, Boston, USA, 6 pp.

Moerman, E.; H. Groeninger and P. van de Homberg; 1986. *De teelt van stooktomaten*. Proefstation voor Tuinbouw onder Glas, Naaldwijk, The Netherlands, 120 pp. (in Dutch).

Montanari, U., 1973. Networks of constraints: Fundamental properties and applications to picture processing. *Information Science*, vol.7, nr.2, p.95-132.

Moore, R.E., 1966. *Interval analysis*. Prentice-Hall series in automatic computation, Englewood Cliffs, USA, 145 pp.

Moore, R.E., 1979. *Methods and application of interval analysis.* SIAM studies in applied mathematics, Philadelphia, USA, 190 pp.

Nadel, B., 1989. Constraint satisfaction algorithms. *Computational Intelligence*, vol.5, nr.4, p.188-224.

Nadel, B., 1990. Representation Selection for Constraint Satisfaction: A Case Study Using n-Queens. *IEEE Expert*, vol.5, nr.3, p.16-24.

Nederhoff, E.M., 1994. *Effects of CO_2 concentration on photosynthesis, transpiration and production of greenhouse fruit vegetable crops.* PhD dissertation Wageningen Agricultural University, The Netherlands, 213 pp.

Nederhoff, E.M., 1995. Techniques of CO_2 enrichment. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach.* Wageningen Pers, Wageningen, The Netherlands, p.195-201.

Nederhoff, E.M.; A.N.M. De Koning and A.A. Rijsdijk, 1992. Leaf deformation and fruit production of glasshouse grown tomato (Lycopersicon esculentum Mill.) as affected by CO2, plant density and pruning. *Journal of horticultural science*, vol.67, nr.3, p.411-420.

Newell, A., 1982. The knowledge level. *Artificial Intelligence*, vol.18, p.87-127.

Norvig, P., 1992. *Paradigms of artificial intelligence programming: case studies in Common LISP*. Morgan Kaufmann Publishers, Los Altos, CA, USA,946 pp.

Older, W.J., and F. Benhamou, 1993. Programming in CLP(BNR). In: Kanellakis P., J.L. Lassez and V. Saraswat (eds.). *Proceedings of first workshop on principles and practice of constraint programming (PPCP'93)*, p.228-238

Older, W.J. and A. Vellino, 1993. Constraint Arithmetic on Real Intervals. In: Benhamou, F. and A. Colmerauer (eds.), *Constraint Logic Programming: Selected Research.* MIT Press, Cambridge, MA, USA, p.175-195.

Paltrinieri, M., 1994. Some Remarks on the Design of Constraint Satisfaction Problems, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, vol.874, p.299-311.

Pearl, J., 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann,San Mateo, CA, USA,552 pp.

Pease, C.M. and J.J. Bull, 1992. Is science logical? *Bioscience* vol.42, p.293-298.

Plant, R.E. and N.D. Stone, 1991.*Knowledge-based systems in agriculture*. McGraw-Hill, New York, USA, 363 pp.

Prosser, P., 1993. Hybrid algorithms for the constraint satisfaction problem. *Computational intelligence*, vol.9, nr.3, p.268-299.

Puget, J.F. and M. Leconte, 1995. Beyond the glass box: constraints as objects. *Proceedings of the International Symposium on Logic Programming*, 4-7 December 1995, MIT Press, Boston USA. p.513-527.

Rich, E. and K. Knight; 1991. *Artificial intelligence.* Second edition. McGraw-Hill, New York, USA, 621 pp.

Rose, M.A., 1994. Oscillatory transpiration may complicate stomatal conductance of gas-exchange measurements. *HortScience*, vol.29, pp.693-694.

Rumbaugh, J.; M. Blaha; W. Premerlani; F. Eddy and W. Lorensen; 1991. *Object-oriented modelling and design*. Prentice-Hall, London, UK 500 pp.

Schmidt, D.C.; 1995. Using design patterns to develop reusable object-oriented communication software. *Communications of the ACM*, vol.38, nr.10, p.65-74.

Schmidt, U., 1996. Greenhouse climate control with a combined model of greenhouse and plant using online measurements of leaf temperature and transpiration. *Acta Horticulturae*, vol.406, p.89-98.

Schotman, P.J., 1989. *Expertsysteem voor de klimaatbesturing in de tomatenteelt.* Master of Science thesis, Department of Horticulture/Department of Computer Science, Wageningen Agricultural University, Wageningen, The Netherlands. (in Dutch)

Schreiber, G.; B. Wielinga and J. Breuker, 1993. *KADS : a principled approach to knowledge-based system development*. Academic Press, London, UK, 457 pp.

Seginer, I., 1991. Optimal greenhouse temperature trajectories for a multi-state-variable tomato model. *Proceedings of the 1st IFAC/ISHS workshop on mathematical and control applications in agriculture and horticulture.* Pergamon press, Oxford, UK. p.73-79.

Seginer, I., 1996. Optimal control of the greenhouse environment: an overview. *Acta Horticulturae*, vol.406, p.191-201.

Seginer, I.; A. Angel; S. Gal and D. Kantz, 1986. Optimal CO_2 enrichment strategy for greenhouses: a simulation study. *Journal of agricultural engineering research*, vol.34, p.285-304.

Seginer, I. and R.W. McClendon, 1992. Methods for optimal control of the greenhouse environment. Transactions of the ASAE, 35, 1299-1307.

Seginer, I. and A. Sher, 1993. Optimal greenhouse temperature trajectories for a multi-state-variable tomato model. In: Hashimoto Y., G.P.A. Bot; W. Day; H.J. Tantau and H. Nonami (eds.) *The Computerized Greenhouse: Automatic Control Application in Plant Production.* Academic Press, New York, USA, p.153-172.

Simon, H.A., 1965. *The shape of automation for men and management.* Harper & Row, New York, USA, 111 pp.

Simon, H.A., 1977. *The new science of management decision.* Revised edition. Harper & Row, New York, USA, 175 pp.

Simonis, H. and N. Beldiceanu, 1995. *The CHIP System.* COSYTEC technical report.

Simonis, H. and T. Cornelissen, 1995. Modelling producer/consumer constraints. *Lecture notes in computer science.* Springer, Berlin, Germany, vol.976, p.449-462.

Sinclair, T., and N.G. Seligman, 1996. Emerging maturity or arrested development in crop modelling. *Quantitative approaches in systems analysis*, AB-DLO, Wageningen The Netherlands, vol.6, p.93-103.

Spaan, J.H. and J.D. Van der Ploeg, 1992. *Toppers en tuinders, bedrijfsstijlen in de glastuinbouw: een verkenning.* Wageningen Agricultural University, Wageningen, The Netherlands, 103 pp. (in Dutch)

Spitters, C.J.T.; H.A.M. Tousaint and D.W.G. Van Kraalingen, 1989. A simple and universal crop growth simulator: SUCROS-87. In: Rabbinge, R. S.A. Ward and H.H. Van Laar (eds.) *Simulation and systems management in crop protection*. Pudoc, Wageningen, The Netherlands, p.147-181.

Stanghellini, C., 1987. *Transpiration of greenhouse crops.* PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 150 pp.

Steele, G.L., 1980. *The definition and implementation of a computer programming language based on constraints.* PhD Thesis, Massachusetts Institute of Technology, Boston, USA.

Steele, G.L., 1990. *Common Lisp: the language, second edition*. Digital Press, Bedford, MA, USA, 1029 pp.

Stefik, M. and D.G. Bobrow, 1986. Object oriented programming: themes and variations. *AI Magazine*, vol.6, nr.4, p.40-62.

Takakura, T., 1975. Plant growth optimization using a small computer. *Acta Horticulturae*, vol.46, p.147-151.

Tantau, H.J., 1985. Greenhouse climate control using mathematical models. *Acta Horticulturae*. Vol.174, p.449-459.

Tap, R.F., in preparation. *Optimal control of greenhouse tomato production*. PhD thesis in preparation, Department of agricultural engineering and physics, Wageningen Agricultural University, Wageningen, The Netherlands.

Tap, R.F.; L.G. van Willigenburg; G. van Straten and E.J. van Henten, 1993. Optimal control of greenhouse climate: computation of the influence of fast and slow dynamics. In: *Proceedings of IFAC 1993 world congress*, vol.10, p.321-324.

Tap, R.F.; L.G. van Willigenburg and G. van Straten, 1996. Experimental results fo receding horizon optimal control of greenhouse climate. *Acta Horticulturae*, vol.406, p.229-238.

Tchamitchian, M.; L.G. van Willigenburg and G. van Straten, 1993. Optimal control applied to tomato crop production in a greenhouse. In: *Proceedings of the european control conference*, vol.3, p.1348-1352.

Thornley, J.H.M. and I.R. Johnson, 1990. *Plant and crop modelling, a mathematical approach to plant and crop physiology*. Clarendon press, Oxford, UK, 669 pp.

Travé-Massuyès, L., 1995. Qualitative reasoning over time: history and current prospects. *Knowledge engineering review*, vol.7, nr.1, p.1-18.

Travé-Massuyès, L. and R. Milne, 1995. Application oriented qualitative reasoning. *Knowledge Engineering Review*. vol.10, nr.2, p.181-204.

Tsang, E., 1993. *Foundations of constraint satisfaction*. Academic Press, London, UK, 421 pp.

Udink ten Cate, A.J., 1983. *Modelling and (adaptive) control of greenhouse climates*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands. 159 pp.

Udink ten Cate, A.J., 1987. Analysis and synthesis of greenhouse climate controllers. In: "Computer applications in agricultural environments.", Eds: Clark, J.A.; Gregson, K. and R.A. Saffell, Butterworths, London, 1-19.

Udink ten Cate, A.J.; G.P.A. Bot and J.J. van Dixhoorn, 1978. Computer control of greenhouse climates. *Acta Horticulturae*, vol.87, p.265-272.

Van der Braak, N.J., 1995. Heating equipment. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, p.171-179.

Van der Maas, A.A., 1991. Teeltbegeleiding in de glastuinbouw: ondersteuning bij de besluitvorming voor gewasbescherming. *Agro-informaticareeks*, vol.5, p.52-59. (in Dutch)

Van Henten, E.J., 1994. *Greenhouse climate control: an optimal control approach*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands 329 pp.

Van Henten, E.J. and J. Bontsema, 1991. Optimal control of greenhouse climate. *Proceedings of the 1st IFAC/ISHS workshop on mathematical and control applications in agriculture and horticulture*. Pergamon press, Oxford, UK. p.27-32.

Van Hentenryck, P., 1998. *The OPL Optimization Programming Language*. The MIT Press, Cambridge, Ma, USA, 255pp.

Van Hentenryck, P.; D.A. McAllister and D. Kapur, 1997a. Solving polynomial systems using a branch and prune approach. *Siam journal on numerical analysis*, vol.34, nr.2 p.797-827.

Van Hentenryck, P.; L. Michel and Y Deville, 1997b. *Numerica: a modelling language for global optimization*. MIT Press, Cambridge, MA, USA, pp. 210.

Van Hentenryck, P. and V. Saraswat, 1997. Constraint programming, strategic directions. *Constraints: an international journal*, vol.2, nr.1, p.7-33.

Van Ieperen, W., 1996. *Consequences of diurnal variation in salinity on water relations and yield of tomato*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 176 pp.

Van Ieperen, W and H. Madery, 1994. A new method to measure plant water uptake and transpiration simultaneously. *Journal of experimental botany*, vol.45, p.51-60.

Van Manson, C. and T. Rieswijk, 1995. Procesbeheersing in de glastuinbouw. *Agro Informatica*, vol.8, nr.4, p.7-10. (in Dutch)

Van Meurs, W.T.M., 1995. Current implementation of hardware and software. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, p.231-240.

Van Roermund, H.J.W., 1995. *Understanding biological control of greenhouse Whitefly with the parasitoid Encarsia Formosa. From individual behavior to population dynamics.* PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 243 pp.

Van Straten, G. and H. Challa, 1995. Greenhouse climate control systems. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, p.249-261.

Van Vliet, H., 1993. *Software engineering, principles and practice*. John Wiley and Sons, Chichester, UK. 558 pp.

Waaijenberg, D., 1995. Construction. In: Bakker J.C.; G.P.A. Bot; H. Challa and N.J. Van der Braak (eds.) *Greenhouse climate control: an integrated approach*. Wageningen Pers, Wageningen, The Netherlands, p.162-170.

Wallace, R.J., 1993. Why AC-3 is almost always better than AC-4 for establishing arc-consistency in CSP's. *Proceedings IJCAI-93*, vol.1, p.239-240.

Waltz, D.,1975. Understanding line drawings of scenes with shadows. In: Winston, P.H. (ed.) *The psychology of computer vision*. McGrawHill, New York, USA.

Waterman, D.A., 1986. *A guide to expert systems*. Addison Wesley, Workingham, UK, 419 pp.

Winston, P.H., 1992. *Artificial intelligence*. Third edition. Addison Wesley, Workingham, UK, 737 pp.

Young, L.F., 1984. A corporate strategy for decision support systems. *Journal of information systems management*, vol.1, no.1 (Winter, 1984).

Young, P.C.; M.A. Behzadi; C.L. Wang and A. Chotai, 1987. Direct digital and adaptive control by input-output state variable feedback pole assignment. *International journal of control*, vol.46, p.1867-1881.

Young, P.C. and M.J. Lees, 1996. Simplicity out of complexity in glasshouse climate modelling. *Acta Horticulturae*, vol.406, p.15-28.

Young, P.C.; S. Parkinson and M.J. Lees, 1995. Simplicity out of complexity in environmental modelling: Occam's razor revisited. *Journal of applied statistics*, vol.23, nr.2-3, p.165-210.

Zadoks, J.C. and R.D. Schein, 1979. *Epidemiology and plant disease management*. Oxford University press, New York, USA, 427 pp.

Zeigler, B.P., 1976. *Theory of modelling and simulation*. Wiley, New York, USA, 435 pp.

Zeigler, B.P., 1984. *Multi-facetted modelling and discrete event simulation*. Academic Press, London, UK, 372 pp.

Zeleny M., 1974. *Linear multi-objective programming.*, Springer-Verlag, Berlin, Germany, 563 pp.

Ziggers, G.W., 1993. *Agrarisch ondernemerschap in een bedrijfskundig perspectief*. PhD thesis Wageningen Agricultural University, Wageningen, The Netherlands, 156 pp. (in Dutch)

APPENDICES

APPENDIX 1. BLOSSOM-END ROT AS AN EXAMPLE OF MODELLING CROP PROBLEMS

1.1 Introduction

In chapter three it has been concluded that much of the knowledge in tomato production, particularly knowledge about disorders and diseases, is qualitative and informal. To use this knowledge in a computerized support system it must first be formalized and represented in a suitable format. Likewise, the resulting model should be thoroughly tested using proper modelling techniques. When the model is finally available, it could for instance be integrated in a computerized support system that carries out a monitoring task.

In this section Blossom-end rot (BER) will be modelled using a combined qualitative–quantitative approach. The model simulates the 'risk of getting BER' based on environmental input. The Blossom-end rot problem has been taken as an example because: *i*) it is important economically, *ii*) it may be difficult to prevent, and, *iii*) there exists a relatively large body of literature about BER (although no attempt to model BER has yet been carried out[202]).

The combined qualitative–quantitative approach used here has been chosen for practical reasons. It is different from the purely numerical approaches normally used in crop modelling (*e.g.* Thornley and Johnson, 1990; Bakker *et al.*, 1995). The idea behind it was to bring together the currently available knowledge about BER (including its underlying processes) without carrying out additional, time-consuming and costly experimentation. Since this knowledge is partly qualitative and partly quantitative, a combined approach was considered necessary.

The model should eventually be usable at commercial nurseries. This requirement precludes the use of both elaborate measuring equipment and complex identification procedures since these are not available, respectively not practicable. The model should have the following characteristics:

1. capture the main causal or influence relations,
2. be understandable for a grower, and,
3. be generally applicable, that is, the model should contain parameters such that it can be calibrated for site- and situation-specific characteristics.

In this section the aim is not to develop a full-fledged model in which the latest and most detailed knowledge about BER has been brought together in an implementation that can readily be used at commercial nurseries. Here, only a basic version of

---

[202] The study of Aikman and Houter (1990) is interesting in this respect since an attempt has there been made to model the calcium supply to the leaves.

such model will be shown. This version is specifically tuned to its foreseen role in a constraint reasoning setting described in the second part of this thesis (sections 6.5.2.2, 6.6.1, 7.4.4 and 8.6).

The structure of the remainder of this appendix is as follows: first, a presentation of the knowledge on BER will be given. Second, based on this body of knowledge, a conceptual model for the 'risk of getting BER' will be constructed. Finally, this model will be implemented by means of constraint reasoning and was used in the experiments in chapter eight.

1.2 Knowledge about BER

Rotting at the distal end of a tomato fruit can be a result of changes in the permeability of the cell membranes (*i.e.* leakage through the plasma membrane or tonoplast) in the distal tissue of a tomato fruit (*e.g.* Spurr, 1959; Van Goor, 1968). This problem, commonly called Blossom-end rot, is an important physiological disorder in tomato and sweet pepper. BER renders the tomato or sweet pepper fruits unsellable, and, because of the economic consequences, a grower considers this a high priority problem. Unfortunately, Blossom-end rot still periodically occurs, especially under summer conditions.

In order to explain the mechanisms behind the occurrence of BER much research has been carried out during the past 30 years. As might be expected, the early investigations (*e.g.* Spurr, 1959; Van Goor, 1968) were explorative in their nature, and related environmental, shoot and root conditions to the incidence of BER. Later studies focused more on the underlying mechanisms.

The 'cause' of BER is generally considered to be a local calcium deficiency in the distal part of the fruit. It is a result of poor distribution of calcium rather than an overall deficiency in the plant or even in the fruit (*e.g.* Bradfield and Guttridge, 1984). Since it is well known that calcium can only be transported (in the plant) through the xylem network (and other, related, apoplastic pathways), much research w.r.t. Blossom-end rot concentrated on Ca^{2+} transport and distribution in relation to the xylem network. For instance, studies of Ho and his co-workers (*e.g.* Belda and Ho, 1993; Ho *et al.*, 1993) showed that the development of the vascular bundle network during the early developmental stages of a tomato fruit, and, the partitioning of Ca^{2+} over different calcium compounds in distinct parts of a berry differ under varying environmental conditions (*i.e.* in this case: varying electrical conductivity).

Knowing that a local calcium deficiency in the distal part of the fruit may be considered the ultimate cause of BER is interesting. However, more practical from a grower's point of view, is knowing how this deficiency can come about and what he can do to prevent it.

When discussing the shortage of calcium in the distal part of a tomato berry both supply and demand processes should be taken in consideration. Therefore, in the subsections below, the processes that determine the calcium demand as well as those that play a role in the calcium supply are discussed.

1.2.1 Demand for calcium

With respect to fruit growth one should discriminate between fresh and dry weight. The volumetric growth of a tomato berry mainly depends upon water import since the water content of a mature fruit is approximately 94%. The dry matter increase of a tomato berry is the result of import of assimilates and nutrients and subsequent metabolic processes. The environmental temperature, the availability of assimilates and the developmental stage of the fruit determine for a large part the rate of these metabolic processes.

During fruit growth, calcium is used as a structural composite in cell membrane, cell wall and xylem vessel tissue. The calcium import rate should keep pace with the rate of fruit growth, since the buffer for calcium in the growing tissue of the berry is limited (Ho, personal communication).

The growth of a tomato fruit varies with its developmental stage and depends, moreover, on the assimilate availability, which, in turn, depends on supply and competition by other fruits (*e.g.* De Koning, 1994).

Tomato fruits grow fastest in the third and fourth week after anthesis (*e.g.* Ho *et al.*, 1987; Bertin and Heuvelink, 1993), and the demand for calcium is consequently the highest (*e.g.* Ho *et al.*, 1993). Since a tomato plant produces one truss a week, there are always two trusses on a plant that have a high calcium demand.

Concludingly, the growth (and thus the demand for calcium) of an individual berry is high when: 1. the assimilate production is high, 2. the temperature is high, 3. the fruitload is low, and 4. the berry is in a fast-growing developmental stage.

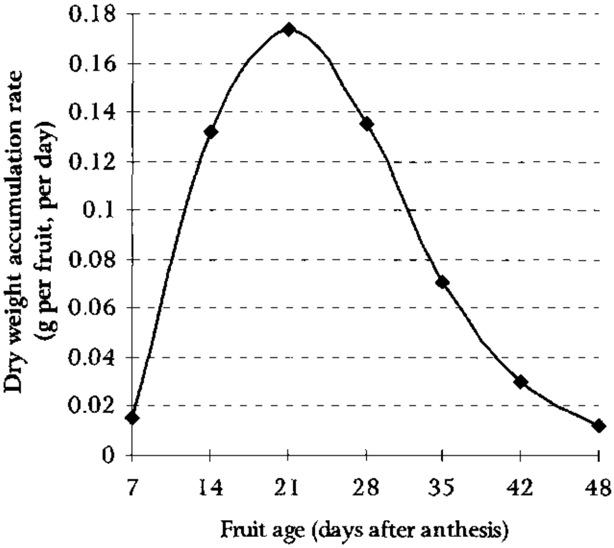


Figure 1-1. Dry matter accumulation rate over time (after Ho at al., 1987)

While the first three factors vary over the season as well as on a day-by-day basis, the fourth factor is to a large extent autonomous, that is, fruits go through stages of varying relative growth rates. In terms of Figure 1-1: the shape of the curve reflects the growth rate of the berry on the basis of its developmental stage; the height of the top of the curve depends upon the other three factors.

1.2.2 Supply of calcium

Much of the research devoted to BER investigates the supply of calcium to the distal part of the berry. Investigations vary from the influence of the root environment (*e.g.* Ehret and Ho, 1986b; Adams and Ho, 1989) to the effects of humidity (*e.g.* Bradfield and Guttridge, 1984; Adams and Holder, 1992). Regarding the translocation of calcium to the distal part of the berry three main processes can be distinguished: the uptake of calcium, the transport of calcium from the roots to the fruit and the transport of calcium within the berry.

1.2.2.1 Calcium uptake

The uptake of nutrients is an energy demanding process that takes place in the endodermis of a root tip. Calcium, like many other ions, is actively loaded by the endodermis cells and unloaded in the apoplastic tissue (xylem vessels) of the root. The uptake of calcium depends upon the following:

- *Root activity*. Here both temperature and assimilate availability play a role. Under normal production conditions both factors are not expected to be the limiting elements for sufficient nutrient uptake.
- *Competition among nutrients*. The loading and unloading pathways for different nutrients partly coincide. It is well known that calcium and potassium (K) use the same mechanism and, consequently, the K/Ca content ratio in the nutrient solution is important for the amount of calcium that is taken up. In the uptake of calcium accompanying anions like phosphate (PO_4^{3-}) also play a role. Under modern production conditions in which growers grow their tomato crop in artificial media like rockwool, the nutrient content can be accurately controlled and should under normal circumstances not be problematic for sufficient uptake of calcium.
- *Age and cultivar differences*. Ho *et al.* (1993) found that plant age and cultivar differences play a role in the amount of calcium taken up by the roots. The uptake of calcium by older plants of salt-tolerant cultivars like Spectra and Calypso deteriorates more compared to older plants of the salt-sensitive cultivar Counter.

The role of the electro-conductivity (EC) in the root environment in relation to the uptake of calcium *per se* is not yet clear (Ho *et al.*, 1995). Tomatoes are, compared to other greenhouse crops like cucumber, relatively insensitive to higher (5 - 7 $mS{\cdot}cm^{-1}$) EC values (Ho and Adams, 1994). The main influence of the electro-conductivity in the root environment will be through the water balance in a tomato plant (*e.g.* Van Ieperen, 1996). A related issue is the relation between calcium uptake and water uptake under varying environmental conditions. Ehret and Ho (1986a) found that the water uptake was reduced much more prominently than the calcium uptake if the relative humidity (r.h.) was changed from a low r.h. (60%) to a high r.h. (90%). They also found that the calcium uptake reduced much more prominently than the water uptake under varying EC's (2 $mS \cdot cm^{-1}$ versus 17 $mS \cdot cm^{-1}$). Their results w.r.t. differences in relative humidity suggest that the calcium concentration in the xylem sap is lower when the water uptake rate is higher, *i.e.* the xylem sap becomes more diluted for calcium when the water uptake conditions improve. In other words, uptake of water and calcium are to some extent independent processes.

1.2.2.2 Transport of calcium to the fruit

Calcium is transported from the roots to the shoot through the xylem. The main driving force in the xylem network is mass flow, that is, calcium and other nutrients flow upward with the water flow through the xylem vessels in stem. The water flow through the plant (and through the xylem in particular) is a result of the overall water uptake, (re-)distribution and transpiration processes. The water balance between the organs (leaves, roots, fruit, etc.) in the tomato plant involves a series of complicated interactions of intra- and inter-organ processes. In-depth coverage of all the issues involved exceeds the goal this section, therefore, only the most important issues for calcium allocation to the fruit will be discussed.

Since the water flow through the xylem to the berry and the calcium concentration in the xylem[203] determine its calcium import, the contribution of the water supply though the phloem should be considered as well. Ho *et al.* (1987) calculated based on accumulated ratio's of ions that 90% of the water enters the fruit through the phloem.

The two principal variables that determine the xylem sap import of an organ are the organ's water potential and the resistance of the xylem pathway to the organ[204]. For reasons of simplicity only the leaves and the fruit are considered here.

It can be concluded that the resistance in the water transport pathway to an individual fruit is much larger than that to a leaf. This is mainly due to the resistance of the pedicel (which connects the fruit with the truss).

The water potential of the leaves may vary considerably due to large variations in leaf transpiration. The main environmental factors that determine transpiration are the light intensity and the humidity of the greenhouse environment, since both fac-

---

[203] High transpiration rates will lead to high flow rates in the xylem, and - given constant calcium uptake - will consequently give rise to lower calcium concentration in the xylem.

[204] Water flow is expressed in $m^3 \cdot s^{-1}$, the water potentials in a plant are usually expressed in MPa (which is a negative value), the resistance is expressed in $MPa \cdot m^{-3} \cdot s$.

tors vary considerably during the course of day, transpiration rates also show large differences within a day. Because tomato berries hardly transpire[205], environmental factors do not the influence the berry's water potential directly. Variations in the water potential of a berry are much less pronounced and to a large extent follow the overall water potential of the plant.

Finally, differences between day and night may also play a role in the overall transport of calcium to the fruit. Unfortunately, the relative contributions of day and night-time calcium import are difficult to determine since various factors come into play simultaneously. Among others: differences between day-time and night-time transpiration (and the resulting mass flow) and differences in the calcium concentration of the xylem sap (and the related calcium unloading rate). Furthermore, the relation between the plant's root pressure (root activity) and the night-time transpiration in relation to the water potential of the leaves and the trusses will also be of importance. To what extent high transpiration rates during the night have a negative impact is not straightforward. The transpiration rates during the night are approximately ten times lower than the transpiration rates during the day (Van Ieperen, 1996). They depend in the first place on the water vapour pressure differences in- and outside the (closed) stomata. To promote upward calcium transport some level of transpiration will be advantageous, however it is not known at what level the nighttime transpiration rate results in a reduction of the transport of calcium to the fruit.

Based on the above it can be concluded that the plant's transpiration rate may be considered the main and most varying factor with respect to calcium transport to the berry. High transpiration rates (especially during the day) will lead to relatively low water potentials in the leaves and thus to a dominant flow of the xylem sap in the direction of the leaves as opposed to the fruit.

1.2.2.3 Intra-fruit calcium transport

A final factor in the occurrence of BER is the transport of calcium to the blossom end of the berry. It has been found that role of calcium is at least two-fold. First, as already stated calcium is needed as a structural composite in the cell wall and cell membrane. Second, calcium influences the lignification of the xylem vessels and thus promotes the conductivity of the individual vessel (Ho *et al.*, 1993). Moreover, calcium availability during berry growth influences the number of xylem bundles per surface area (Belda and Ho, 1993). Both the number of xylem bundles and their conductivity determine the intra-fruit transport capacity for calcium to the distal part of the berry. During periods of rapid volumetric growth, a limited transport capacity will negatively influence the calcium availability for cell expansion (cell membrane and cell wall tissue) in the distal part of the berry .

---

[205] The role of the calyx should be mentioned since it transpires and thus stimulates mass flow to the berry. Ehret and Ho (1986c) found that removal of the calyx indeed promotes BER.

1.2.3 Other influences

There are two influences on the risk of getting BER that haven't yet been discussed.

The first one is the influence of the cultivar. Adams and Ho (1992) found that salt-resistant cultivars are more sensitive to BER than cultivars that do not have this characteristic. It is believed (Ho, personal communication) that the salt-resistant cultivars are able to keep up volumetric fruit growth (resulting in a higher average fruit weight), while the calcium uptake lags behind.

The second one is the influence of sharp weather changes (particularly, when a sudden burst of bright weather follows a period of dull weather) on BER sensitivity. A period of dull weather has its effect on the morphology of the root and shoot. The root tips have been adapted to the dull weather conditions and, consequently, the nutrient uptake capacity has dropped (dull weather means less growth thus less nutrients are needed). At the shoot side, the leaves have also adapted. During dull weather leaves generally become larger, in addition, morphological changes within developing leaves (*e.g.* higher stomata density) could also be a factor. Consequently, the plant's transpiration capacity increases. After a weather change: *i*) a relative larger fraction of the xylem water flows to the leaves, while *ii*) the root tips may not be able to keep up with the much higher calcium demand.

1.2.4 Observations in practice

BER mainly occurs during summer conditions. Interviews with growers and an extension service specialist revealed that keeping sufficient fruitload is considered important to prevent BER[206]. Moreover, growers with BER are advised to keep the affected fruit on the plant, since removing them may promote BER on younger fruits. Both observations relate to an increase in the assimilate supply for the individual fruit (and thus the demand for calcium) and are in line with our knowledge on BER (see above).

Discussions with growers and researchers revealed that BER is also frequently encountered after a (local) malfunction of the irrigation system. During summer conditions a breakdown of one day may result in the occurrence of BER on two trusses. If during such a breakdown the plants get no water, the rockwool slabs will quickly run dry. As a result the amount of xylem water that flows to the fruit will strongly diminish (and consequently the amount of calcium).

---

[206] Unfortunately, keeping a sufficient fruitload can be problematic in the early summer (end of June, start of July) because of an increased ripening due to the - on average - higher temperatures.

1.2.5 Conclusions

The main influences on the risk of getting BER have been summarized in Figure 1-2.

In this figure the risk of getting BER is expressed at the individual fruit level. On the demand side fruit growth is the principal process that determines the need for calcium. The rate of fruit growth depends on environmental factors (mainly temperature) and on assimilate availability, which, in turn, depends on crop photosynthesis and the inter-fruit competition.

On the supply side the two main processes that influence the supply of calcium to the individual fruit are the transpiration and the calcium uptake. Contrary to the demand side, no process has been identified that actively regulates the supply of calcium in or to the individual fruit, indeed, mass flow in the xylem network can be seen as the dominant driving force of the calcium supply to the fruit. Transpiration is the cause of this mass flow. With respect to the influence of transpiration on the amount of calcium supplied to the fruit, one may in theory assume that there is a kind of "optimum" (therefore, in Figure 1-2, its influence has been marked with a +/- sign). This so-called optimum depends amongst others on the state of the crop and is likely to vary over the day (at least there is a difference between day and night).

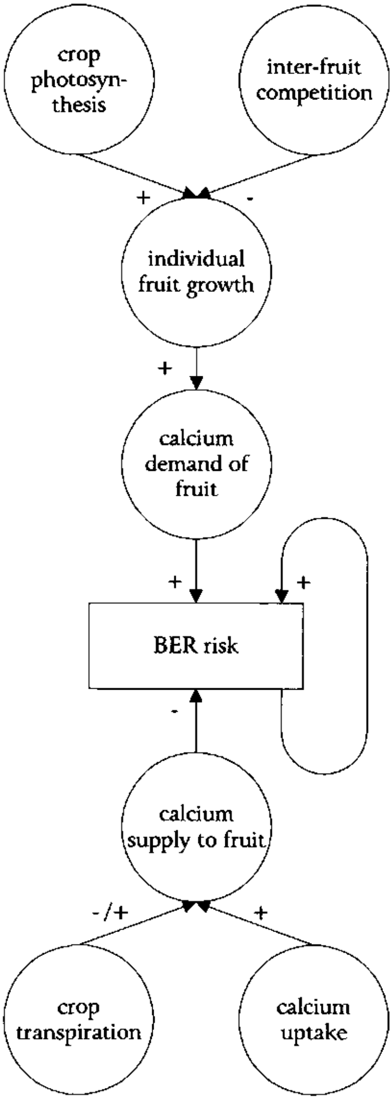


Figure 1-2 Influences on the risk of getting BER, (+) means positive influence, (-) negative influence.

The influence of cultivar susceptibility and morphological adaptations, as discussed in section 1.2.3, are not shown since both influences are considered internal to the calcium uptake, respectively the transpiration processes.

Finally, the risk of getting BER is considered to be a fruit characteristic in which history plays a role. Increase of the risk of getting BER can be thought of as a gradual build up of a Calcium deficit over several days. This is depicted through the feedback loop in Figure 1-2.

1.3 Constraint model of the 'risk of getting BER'

1.3.1 Model description

The knowledge described above has been cast in a constraint model[207]. In this model the constraint types described in sections 7.3.7 and 7.4 have been used; furthermore the model has been placed in a time hierarchy as discussed in section 7.3.2. The model consists of a small number of qualitative variables and constraints. The constraints C-qual-1 to C-qual-5 in Figure 1-3 describe the main influences on the risk of getting BER. The link between this qualitative model and other - quantitative - models or data sources is achieved by means of mapping constraints (C-map-1 through C-map-4). These mapping constraints connect qualitative variables with their quantitative counterparts (and vice versa). The quantitative variables are either part of quantitative models or can be directly obtained from the greenhouse environment (*i.e.* they can be measured respectively set by the grower).

The model simulates the risk of getting BER on a daily basis. Since it is assumed that history plays a role in the risk of getting BER, feedback from the previous day has been built into the model.

Compared to the description in section 1.2.5, the constraint model contains the following simplifications:

- The constraint model describes the BER risk at the crop level while in section 1.2.5 the BER risk at individual fruit level was considered. This simplification[208] is realized by considering the fastest growing fruit only (*i.e.* the variable "Fruit growth (quantitative)" in Figure 1-3).
- Since it is believed that the calcium demand is highly correlated with fruit growth, the variable "Calcium demand" is directly connected with the variable "Fruit growth (quantitative)" through the mapping constraint C-map-1. Conceptually, this mapping constraint involves more than a simple mapping procedure, it also represents a transformation from one concept to another. Moreover, calibration of the C-map-1 constraint is necessary to ensure that calcium supply and calcium demand relate logically to one another (in the constraint C-qual-2).

---

[207] One should keep in mind that in a constraint model domain reduction is always bidirectional (see section 8.6 for an example).

[208] It is clear that through this simplification mass-balance laws do not hold.

- Calcium uptake is assumed to be directly related to the electrical conductivity (EC) of the root environment, therefore, in the constraint C-map-4, the quantitative variable "EC" maps directly to the qualitative variable "Calcium uptake". Other influencing factors w.r.t. the environment and condition of the roots are ignored. Since the EC is a slowly changing variable, distinction between day- and nighttime has not been built into the model (consequently, the calcium uptake is also assumed to be the same over the day).
- Since daytime transpiration is approximately ten times higher than the nighttime transpiration, day- and nighttime calcium supply are modelled separately.

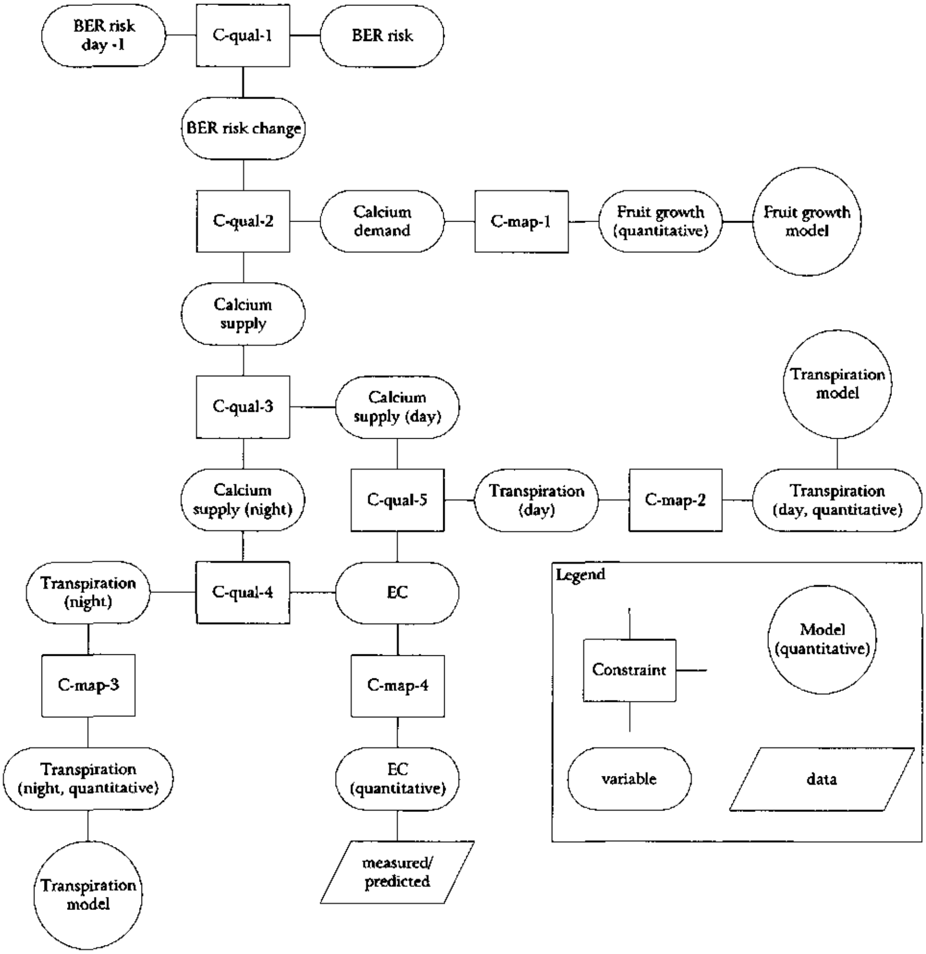


Figure 1-3 Constraint model of the risk of getting BER.

Finally, the meaning of the phrase 'risk of getting BER' may be discussed. In this model the term 'risk' should be seen from the standpoint of a grower. Since the grower is mainly interested in the crop as a whole, he looks at BER from a population perspective (*i.e.* all the fruits of his crop). He is not interested whether fruit *x* on plant *y* in compartment *z* will get BER or not. In a sufficiently tuned model, a very high risk of getting BER should imply that a relatively large number of the most susceptible fruits will get BER (the fruits that are most vulnerable are the ones that grow the fastest). Lower risks of getting BER will show less and less fruits affected. When the risk of betting BER has the value `average` or `normal`, BER should not occur in the greenhouse.

1.3.2 Individual constraint relations

In this section, the qualitative constraints C-qual-1 to C-qual-5 of Figure 1-3 have been stated in a tabular format in which the axis make up the 'input' values and the table cells hold the 'resulting values'.

The model relationships have been built up in such a way that small disturbances in the calcium balance dampen out, that is, a kind of stability has been build into the model. At the same time however, when high risks eventually occur, they are difficult to get rid off.

1.3.2.1 The risk of getting BER (BER risk) as a function of the old value and its change

Table 1-1 describes the relationship between the risk of getting BER, the previous risk value and its change. It can be observed in this table that for both 'input' variables, high values have a more pronounced effect than low values. In essence this relationship suggests that it is more easy to get a high risk than to get rid of it.

Table 1-1 Constraint C-qual-1: BER risk.

	BER risk change						
BER risk$_{day-1}$	*very low*	*low*	*moderately low*	*average*	*moderately high*	*high*	*very high*
very low	very low	low	low	moderately low	average	average	moderately high
low	low	low	moderately low	moderately low	average	moderately high	moderately high
moderately low	low	moderately low	moderately low	average	average	moderately high	high
average	moderately low	moderately low	average	average	moderately high	moderately high	high
moderately high	average	average	average	moderately high	moderately high	high	high
high	average	moderately high	moderately high	moderately high	high	high	very high
very high	moderately high	moderately high	high	high	high	very high	very high

1.3.2.2 The change in the risk of getting BER (BER risk change) as a function of the calcium supply and demand

Table 1-2 describes the relation ship between the change in the risk of getting BER, the supply of calcium and demand of calcium. The table shows that if supply and demand are balanced the change in risk is `average`. The table also shows that differences between supply and demand move away from the average in a less than proportional manner.

Table 1-2 Constraint C-qual-2: BER risk change.

Calcium supply	**Calcium demand**						
	very low	*low*	*moderately low*	*average*	*moderately high*	*high*	*very high*
very low	average	moderately high	moderately high	high	high	very high	very high
low	moderately low	average	moderately high	moderately high	high	high	very high
moderately low	moderately low	moderately low	average	moderately high	moderately high	high	high
average	low	moderately low	moderately low	average	moderately high	moderately high	high
moderately high	low	low	moderately low	moderately low	average	moderately high	moderately high
high	very low	low	low	moderately low	moderately low	average	moderately high
very high	very low	very low	low	low	moderately low	moderately low	average

1.3.2.3 The total calcium supply as a function of the day- and nighttime supply

Table 1-3 describes the relationship between the total calcium supply, the daytime supply and the nighttime supply. The table shows that the total calcium supply is the maximum of the day- and nighttime supply. This relationship implies that compensation over the 24-hour period is possible.

Table 1-3 Constraint C-qual-3: Calcium supply.

nighttime Ca supply	**daytime Ca supply**						
	very low	*low*	*moderately low*	*average*	*moderately high*	*high*	*very high*
very low	very low	low	moderately low	average	moderately high	high	very high
low	low	low	moderately low	average	moderately high	high	very high
moderately low	moderately low	moderately low	moderately low	average	moderately high	high	very high
average	average	average	average	average	moderately high	high	very high
moderately high	moderately high	moderately high	moderately high	moderately high	moderately high	high	very high
high	high	high	high	high	high	high	very high
very high	very high	very high	very high	very high	very high	very high	very high

1.3.2.4 The nighttime and daytime calcium supply as a function of the nighttime, respectively daytime transpiration and the calcium uptake

Table 1-4 and Table 1-5 describe the relationship between the calcium supply, the transpiration and the calcium uptake during night, respectively day. The tables show that the influence of transpiration differs between day and night. During the night is a relatively high transpiration advantageous while during the day a relatively low transpiration brings about the most supply of calcium.

Table 1-4 Constraint C-qual-4: Nighttime calcium supply.

Calcium uptake	**Nighttime transpiration**						
	very low	*low*	*moderately low*	*average*	*moderately high*	*high*	*very high*
very low	very low	low	moderately low	moderately low	moderately low	low	very low
low	low	moderately low	average	average	moderately low	low	low
moderately low	moderately low	average	average	average	average	moderately low	low
average	moderately low	average	moderately high	moderately high	average	average	moderately low
moderately high	average	moderately high	moderately high	moderately high	moderately high	average	moderately low
high	moderately high	moderately high	high	high	moderately high	average	average
very high	moderately high	high	very high	very high	high	moderately high	moderately high

The optimum in the nighttime calcium supply can be argued as follows; if the transpiration rate is very low, the flow of calcium to the upper parts of the plant comes more or less to a standstill. On the other hand when the transpiration is very high, the mass flow behavior of calcium starts to resemble its behavior during daytime.

Table 1-5 Constraint C-qual-5: Daytime calcium supply.

Calcium uptake	**Daytime transpiration**						
	very low	*low*	*moderately low*	*average*	*moderately high*	*high*	*very high*
very low	low	moderately low	moderately low	moderately low	moderately low	low	very low
low	moderately low	moderately low	average	average	average	moderately low	low
moderately low	average	average	average	average	moderately low	low	very low
average	moderately high	moderately high	average	average	moderately low	low	very low
moderately high	high	moderately high	average	moderately low	low	very low	very low
high	moderately high	average	moderately low	low	low	very low	very low
very high	moderately high	average	moderately low	low	very low	very low	very low

1.3.3 Further developments

The model relations described are the central part of model describing the risk of getting BER. Another important component are the mapping constraints C-map-1 to C-map-4. These constraints have not yet been formulated. The development of the mapping constraints C-map-2 and C-map-3 requires data generation by means of experimentation in practice. When the data have been gathered, they can be evaluated by experts in the field (*e.g.* consultants of the extension service and growers), and a mapping distribution can be set up.

The mapping constraint C-map-1 requires simulation with the selected fruit growth model. After extensive simulation under a broad range of conditions, the results can be analyzed and the daily fruit growth distribution can be mapped to the set of qualitative values for the calcium demand. Given the lack of experience with fruit growth simulation, this mapping constraint is likely to require some tuning once the complete model BER risk model is in place.

A mapping distribution between the electro-conductivity (EC) and the calcium uptake for the constraint C-map-4 can be obtained from discussions with growers and extension service consultants.

Finally, the complete model for the risk of getting BER will require extensive testing and tuning in practice. Nevertheless the approach is believed to be generic and suitable for similar cases.

APPENDIX 2. THE `REVISE` METHOD FOR NUMERICAL CONSTRAINTS USING THE INTERVAL NEWTON ALGORITHM

2.1 Introduction

The `revise` method for numerical constraints used in this work is an extension of the interval Newton algorithm presented in Hansen (1992, p.74-75). The extension is better equipped to deal with perturbed equations, which is necessary since every n-ary constraint includes multiple variables that have domains which are initially wide. The algorithm reduces the domains of the variables in the constraint to their minimal interval with respect to the fixed perturbance in the constraint equation, proof of this attribute can be found in Dinkel *et al.*(1988).

2.2 Properties of the algorithm

The algorithm has a number of properties that makes it an excellent candidate to be used in constraint reasoning procedures (Van Hentenryck, 1997a; Van Hentenryck *et al.*, 1997b). The algorithm is a relatively simple and has a proven track record (Van Hentenryck *et al.*, 1997a). Three properties stand out:

1. Every zero of the function $f(x_n)$ w.r.t. x_i for in the initial interval X_0 of x_i will always be found and correctly bounded. In constraint terminology: the algorithm returns the smallest feasible interval domains for the argument variables $x_1 .. x_n$ in constraint c (w.r.t. the fixed perturbance in c). Constraints in the form of eqn. (7) should be rewritten in the form of eqn. (8). A gradient function or the partial derivative of f must be available for every variable in the constraint.

$$c : x_n = g(x_1 \ldots x_{n-1}) \tag{7}$$

$$f(x_1 \ldots x_n) = g(x_1 \ldots x_{n-1}) - x_n = 0 \tag{8}$$

2. If there is no zero for $f(x_n)$ in X_0 then the algorithm will prove this fact in a finite number of iterations. In constraint terminology: inconsistent domains will always be removed.
3. Rapid convergence to the smallest feasible domain for every variable in a constraint.

Proofs of these properties can be found in Hansen[209] (1992) and Dinkel *et al.*(1988). Note that since this algorithm has been embedded in a local consistency checking mechanism, global consistency can only be realized *after* search.

The current representation differs from Hansen (1992, p.74-75) in the following aspects:

- The algorithm is extended in that it is now a triple pass version

  (*i.e.* the `for` ... `do` loop) which differ only in the value taken for x. During the first pass x is taken as the midpoint of X, however, in perturbed problems that exhibit considerable uncertainty the use of the midpoint may not reduce the original interval X at all. Therefore, during the second and third pass x is set to the left and right boundary of X respectively. In general, this triple pass approach is considerably more efficient (*i.e.* on average 5 to 10 times faster) compared to using a double pass approach in which only the boundaries of X are taken. In theory, the boundaries are sufficient to arrive at the minimal interval according to the proof in Dinkel *et al.*, 1988. In non-perturbed cases, the second and third pass will not contribute to the solution because stopping criteria are already met.
- Errors in equations 2.3.4. and 7.2.3. (Hansen 1992, p.9 resp. p.69) have been corrected.

Additional *efficiency* gains are possible and mostly refer to the dependency problem[210]. Equations that suffer from the dependency problem require more function evaluations as opposed to those do not suffer from it (or in which the problem is less severely present (*e.g.* section 7.2.2.2).

The interval Newton method has an advantage over other proposals for constraint reasoning in interval domains (*e.g.* Hyvönen, 1992) which required inversion of the constraint equations for each variable in the constraint. For complicated equations inversion is not always possible; solutions for this problem requires the introduction of 'slack' of intermediate variables. In general, introduction of these 'slack' variables worsens the dependency problem. The interval Newton method requires a gradient functions which in general are not difficult to determine.

---

[209] Based on Moore (1966), Hanson (1969), and Hansen and Greenberg (1983).

[210] The dependency problem does not impose efficacy problems (*see* Dinkel *et al.*, 1988).

The method uses the following procedure:

$$\text{Newton}(F,F,I) = \bigcap_{i=0}^{\infty} I_i$$

$$I_{i=0} = \text{initial interval } I$$

$$I_{i+1} = \text{N}(F,F,I_i) = I_i \cap \left\{ \text{center}(\text{I}_\text{i}) - \frac{\text{F}(\text{center}(I_i))}{\text{F}'(I_i)} \right\}$$

F is the original constraint equation, F' is its derivative.

In the description of the interval Newton algorithm, the conventions explained in Table 2-1 are used.

Table 2-1 Notation below.

variableName	Variable names are stated in italics, with the first word of the variable in lower case, while the first letter of every additional word is stated in upper case. Variables ending with a question mark are Boolean.
PROCEDURENAME () or *PROCEDURENAME ()*	Procedures are stated in smallcaps; the first letter of each new word in the procedure name is stated in upper case. Procedure names that are stated in italics are either plain accessors[211] or simple procedures that are considered to be described sufficiently by their name alone. The argument list of a procedure is given between parenthesis.
`keywords`	Keywords that are part of the 'language' that is used to describe the algorithm are given in the underlined courier font.
`booleans`	Boolean or logical values are in `courier`.
;	A semicolon denotes the end of a statement.
//	The double (forward) slashes denote comment strings.

---

[211] Accessors are procedures that get the value of a slot of an object.

2.3 The main interval Newton algorithm

```
proc REVISE (aNumericalConstraint) // returns a list of constraints
    varListToProcess := VARS (constraint);
    globalChange? := true;
    constraintsNeedRecheckingList := ∅;
    while {  NOT ( EMPTYDOMAIN(varToProcess) ) and varListToProcess ≠ ∅ } do
        varToProcess := POP (varListToProcess);
        INITIALIZE (varToProcess);
        while {  NOT ( EMPTYDOMAIN(varToProcess) ) and globalChange? } do
            globalChange? := false;
            intervalCombinationList := INTERVALCOMBINATIONS (varToProcess, VARS (constraint));
            while intervalCombinationList ≠ ∅ do
                xIntervalValue := POP (intervalCombinationList);
                changeDuringMidpointFlip? := true;
                //  xIntervalValue is an object which also includes other information such as
                //  bookeeping information and the values (beloning to the combination)
                //  for the other variables in the constraint.
                while changeDuringMidpointFlip? do
                    changeDuringMidpointFlip? := false;
                    forall xValue ∈ {MIDPOINT(xIntervalValue), RIGHTBOUND(xIntervalValue), LEFTBOUND(xIntervalValue)} do
                        localChange? := true;
                        while localChange? do
                            localChange? := false;
```

```
resultFunction := APPLY (FUNCTION(aNumericalConstraint), xIntervalValue);
if CHECKSTOPPINGCRITERIA (xIntervalValue, resultFunction)
   BOOKKEEPING(xIntervalValue);
   // Only bookkeeping, because stopping criteria should apply for every midpoint
else
   lastBounds := xIntervalValue;
   resultDerivative := APPLY (DERIVATIVE(aNumericalConstraint), xValue);
   if $0 \notin$ resultDerivative
      xIntervalNew := CALCULATENORMALNEWTON (resultFunction, resultDerivative, xValue);
      xIntervalValue:= CHECKOVERLAP (xIntervalValue, xIntervalNew);
      if xIntervalValue = $\varnothing$
         BOOKKEEPING(xIntervalValue);
         changeDuringMidpointFlip? := false;
      elseif xIntervalValue = "unchanged"
               // For efficiency reasons one may also allow the algorithm to stop when
               // the domain reductions during this step are very small.
         BOOKKEEPING(xIntervalValue);
      else
         changeDuringMidpointFlip? := true;
         localChange? := true;
      endif
   elseif $0 \notin$ resultFunction
      xIntervalNew := CALCULATEEXTENDEDNEWTON (resultFunction, resultDerivative, xIntervalValue,
                                               xValue, intervalCombinationList);
      xIntervalValue:= CHECKOVERLAP (xIntervalValue, xIntervalNew);
```

```
                            if xIntervalValue = ∅
                                BOOKKEEPING(xIntervalValue);
                                changeDuringMidpointFlip? := false;
                            elseif xIntervalValue = "unchanged"
                                BOOKKEEPING(xIntervalValue);
                            else
                                changeDuringMidpointFlip? := true;
                                localChange? := true;
                            endif
                        elseif CALCULATESPLITDETERMININGVALUE (resultFunction, resultDerivative);
                            BOOKKEEPING(xIntervalValue);
                        else
                            xIntervalValue := SPLITANDSTACK (xIntervalValue);
                            changeDuringMidpointFlip? := true;
                            localChange? := true;
                        endif
                    endif
                endwhile        //  localChange?
                if xIntervalValue = ∅
                    "jump out the forall xValue do loop";
                endif
            endfor
        endwhile        //  changeDuringMidpointFlip?
    endwhile        //  intervalCombinationList
    globalChange? := { BOOKKEEPING(varToProcess) or globalChange? };
endwhile        //  globalChange?
```

```
        if {  NOT ( EmptyDomain(varToProcess) ) and RealDomainChange(varToProcess) }
            constraintsNeedRecheckingList := AddConstraints(constraintsNeedRecheckingList, varToProcess);
        endif
    endwhile      //  varListToProcess
    if EmptyDomain(varToProcess)
        NotifyInconsistency ();
    else
        Return(constraintsNeedRecheckingList);
    endif
endproc
```

2.4 Procedures supporting to the main interval Newton algorithm

```
proc INITIALIZE (varToInitialize)
    Backs up original domain, clears "new intervals to be merged" slot, sets logicals: "empty domain", "real doman change" to false.
endproc

proc INTERVALCOMBINATIONS (varToProcess, otherVariables)
    Generates a list of all possible domain combinations.
endproc

proc CHECKSTOPPINGCRITERIA (xIntervalValue, lastBounds, resultFunction)
    Stopping criteria (Hansen, 1992) are: i) interval width and ii) magnitude of the function evaluation compared to their respective error criteria.
endproc

proc BOOKKEEPING(xIntervalValue)
    Collects (un)changed interval for merging (merging will be done at a later step), or, if empty removes the interval (that may have been added in earlier steps of the for loop) from this "list to be merged".
endproc

proc CALCULATENORMALNEWTON (resultFunction, resultDerivative, xValue)
    xValue - resultFunction / resultDerivative
endproc

proc CHECKOVERLAP (xIntervalValue, xIntervalNew)
    Determines and returns the intersection between the two intervals.
endproc

proc CALCULATEEXTENDEDNEWTON (resultFunction, resultDerivative, xIntervalValue,
                              xValue, intervalCombinationList)
    Calculates xValue - resultFunction / resultDerivative using extended interval arithmetic, if the calculation results in two intervals, one will be added to the intervalCombinationList and the other will be returned.
endproc
```

`proc` CalculateSplitDeterminingValue(*resultFunction*, *resultDerivative*)

Determines whether splitting the 'wide' solution interval may be useful (*see* Hansen, 1992, p.73 for a discussion).

`endproc`

`proc` SplitAndStack (*xIntervalValue*, *intervalCombinationList*)

If the solution was too wide, interval splitting may lead to a more narrow solution. One half of the interval will be returned the other will be placed on the *intervalCombinationList*.

`endproc`

`proc` Bookkeeping(*varToProcess*)

The union of all changed and unchanged intervals will be compared to the original domain and whether a real change has occurred is determined. Possibly results in empty domain if all combinations evaluated to empty. Sets the logicals "real domain change" and "empty domain detected".

`endproc`

`proc` AddConstraints(*constraintReturnList*, *varToProcess*)

Checks the *varToProcess* for constraints it participates in and adds them to the *constraintReturnList* if they are not already on this list.

`endproc`

The following procedures are accessors:

Midpoint (), *RightBound* (), *LeftBound* (), *Function* (), and *Derivative* (),

The following procedures are assumed to be sufficiently explained by their name:

Not (), *Pop ()*, *Apply* (), *RealDomainChange* (), *EmptyDomain* (), *NotifyInconsistency* (), and *Return*().

SAMENVATTING

(Summary in Durch)

Dit proefschrift behandelt de geautomatiseerde ondersteuning voor het operationele management van kasgewassen. In dit proefschrift is de tomaat als voorbeeldgewas genomen, omdat het bij aanvang van het onderzoek, dat aan dit proefschrift ten grondslag ligt, het belangrijkste Nederlandse tuinbouwgewas was. De beschreven resultaten zijn echter te veralgemeniseren naar gewassen die qua groeiwijze en teeltmethode op de tomaat lijken (komkommer, paprika, etc.).

Het operationeel management op een tuinbouwbedrijf wordt primair uitgevoerd door de tuinder en is in het kader van dit proefschrift gedefinieerd als de dagelijkse besluitvorming die leidt tot activiteiten welke direct of indirect de groei en ontwikkeling van het gewas beïnvloeden. Binnen het operationeel management beschikt de tuinder over geautomatiseerde ondersteuning in de vorm van procesbesturingssystemen voor de regeling van het kasklimaat en de watergift, inclusief de nutriëntendosering. Deze systemen, in het bijzonder de systemen voor de regeling van het kasklimaat, staan vanwege het geïdentificeerde verbeterpotentieel reeds geruime tijd in de belangstelling van het wetenschappelijk onderzoek. Mogelijkheden tot verbetering vinden onder andere hun oorsprong in het feit dat de huidige regelsystemen geen gewasspecifieke kennis bevatten, d.w.z. kennis die voorspelling en dus optimalisering van de gewasresponse mogelijk maken. Bovendien zijn er in de praktijk grote verschillen in energie-efficiëntie waargenomen, deze verschillen zijn gedeeltelijk te herleiden tot sub-optimale instellingen van het klimaatbesturingssysteem.

Met het doel de geautomatiseerde ondersteuning voor het operationele management te verbeteren is in dit proefschrift een concept ontwikkeld voor een kennisgebaseerd systeem. Met de term 'kennisgebaseerd' wordt aangegeven dat het systeem kennis uit het probleemdomein (d.w.z. de teelt van tomaten in kassen) moet bevatten. Het gaat hier dus om gewasspecifieke kennis (d.w.z. kennis over de tomaat), maar ook om algemene modelkennis binnen de teelt van kasgewassen (bijvoorbeeld processen die de dynamica van het kasklimaat beschrijven, en processen als fotosynthese en transpiratie). Bij de ontwikkeling van dit systeemconcept is met nadruk gezocht naar een balans tussen mogelijkheden die moderne methoden en technieken voor de beslissingsondersteunende systemen kunnen bieden, en hun toepasbaarheid in de dagelijkse tuinbouwpraktijk.

Verbetering van de geautomatiseerde ondersteuning voor het operationele management kan niet zonder een goed inzicht in het probleemdomein. Het probleemdomein staat dan ook centraal in hoofdstuk twee. In dit hoofdstuk wordt de teelt van kasgewassen en in het bijzonder de teelt van de tomaat uit het oogpunt van de operationele besturing beschouwd. Allereerst worden de algemene productiekarakteristieken van de tomatenteelt besproken; hierbij komt naar voren dat er een grote verschillen in teeltomstandigheden waar te nemen valt (d.w.z. variabiliteit binnen één seizoen, tussen opeenvolgende seizoenen, en tussen verschillende productielocaties en bedrijfsuitrusting). Vervolgens wordt de operationele besluitvorming van de tuinder nader geanalyseerd en in het kader van de theorie over besluitvormingsprocessen en beslissingsondersteuning geplaatst. Deze analyse resulteert in een model van de operationele besturing van een tomatengewas. Centraal in dit model staat de "intelligence, design and choice" besluitvormingscyclus van Simon (1977). De "design and choice" fasen in dit model beslaan de vertaalslag van het resultaat van de beoordeling van de gewas- en omgevingssituatie (uit de 'intelligence' fase) naar een verzameling van te implementeren acties. Binnen deze transitieslag moet de tuinder van het gewas- en omgevingsniveau 'afdalen' naar het stuurniveau, hetgeen deze vertaalslag tot een complexe en kennisintensieve activiteit maakt.

> Bijvoorbeeld: de tuinder merkt op dat het gemiddeld aantal vruchten per m^2 te snel achteruitloopt en wil hier iets aan doen. De te snelle daling van het gemiddeld aantal vruchten per m^2 zou hij bijvoorbeeld via vertraging van de rijpingssnelheid van de vruchten kunnen aanpakken. Het vertragen van de rijpingssnelheid kan gerealiseerd worden door een verlaging van de gemiddelde temperatuur. De gemiddelde temperatuur is opgebouwd uit het resultaat van de verwarmings- en ventilatietemperatuur die voor verschillende dagdelen apart door de tuinder worden ingesteld. Kortom: om het probleem van te snelle daling van het gemiddeld aantal vruchten per m^2 aan te pakken, moet de tuinder aanpassingen overwegen op het stuurniveau (bijv. verlaging van de verwarmingstemperatuur gedurende de nacht) die relatief ver van het originele probleem afstaan. Aangezien de geschetste oplossingsstrategie slechts één van de alternatieve oplossingsrichtingen is, moet de tuinder meer van deze oplossingsmogelijkheden in overweging nemen.

Deze complexe vertaalslag van attributen/variabelen op het gewas- en omgevingsniveau naar variabelen op het stuurniveau vormt later in het proefschrift een belangrijk aanknopingspunt voor een nieuw systeemconcept.

In hoofdstuk drie wordt de kennis binnen het probleemdomein nader geanalyseerd. De verschillende kenmerken en representatiemethoden van kennis worden toegelicht en de beschikbare kennisbronnen binnen het probleemdomein worden besproken. Hierbij blijkt, dat de kennis met betrekking tot het ontstaan van fysiologische afwijkingen en de ontwikkeling van plagen en ziekten nauwelijks formeel beschreven is en uitsluitend in kwalitatieve vorm beschikbaar is, en bovendien sterk afhankelijk is van de context waaruit de kennis betrokken is. Om meer inzicht te verkrijgen in de relaties tussen de belangrijkste processen en variabelen binnen het probleemdomein zijn deze aan de hand van verschillende aspecten geordend. Deze structurering laat onder andere zien dat de tijdconstanten van belangrijke processen binnen het probleemdomein sterk uiteenlopen. Tenslotte wordt in hoofdstuk drie ingegaan op specifieke aandachtspunten, waar, bij de verdere kenniselicitatie (≈ kennisvergaring) binnen dit probleemdomein op moet worden gelet.

In hoofdstuk vier komen mogelijkheden voor additionele beslissingsondersteuning aan de orde. Allereerst worden de belangrijkste eigenschappen van de huidige regelsystemen besproken. Het blijkt dat de geleidelijke ontwikkeling van de systemen er mede een oorzaak van is dat de complexiteit van het gebruikersinterface sterk is toegenomen. Tevens valt op dat de huidige systemen nog steeds gebruik maken van (relatief traditionele) heuristische regelmethodieken. Aan de hand van het besluitvormingsmodel van Simon (1977) worden vervolgens de verschillende vormen van beslissingsondersteuning behandeld. Ten aanzien van de ondersteuning van de beoordelingstaak van de tuinder worden ontwikkelingen op het gebied van sensortechnologie en mogelijkheden van modelgebaseerde ondersteuning nader uiteengezet. Het blijkt dat de toepasbaarheid van beide typen van ondersteuning niet triviaal is. Aspecten, waarop de toepassing van deze nieuwe methoden onder andere mis kan gaan, zijn: (onvoldoende) robuustheid van zowel apparatuur als model, representativiteit van een meetlocatie, problematische modeltuning, en de interpretatie van modelresultaten. Ten aanzien van de ondersteuning van de "design and choice" fasen uit de besluitvormingscyclus wordt in dit hoofdstuk een breed spectrum van mogelijke benaderingen uit verschillende vakgebieden bediscussieerd. Uit deze literatuurstudie blijkt dat alle onderzochte benaderingen gekenschetst kunnen worden als ambitieus tot zeer ambitieus omdat zij de besluitvormingstaken van de tuinder geheel of gedeeltelijk overnemen. Dit laatste is opvallend daar in hoofdstuk twee is aangetoond dat er over bepaalde besluitvormingstaken nog te weinig kennis is in de vorm van duidelijk omlijnde besluitvormingsprocedures of eenduidige criteria waartegen een keuze kan worden geëvalueerd. Tenslotte moet worden opgemerkt dat er nauwelijks benaderingen gevonden zijn die onder praktijkomstandigheden zijn getoetst.

Als afsluiting van het analysedeel van dit proefschrift worden in hoofdstuk vijf de eerder beschreven invalshoeken bij elkaar gebracht. Op basis van de eigenschappen van het operationele managementprobleem, de kenmerken van de beschikbare kennis binnen het probleemdomein, en de beschikbare technieken voor additionele beslissingsondersteuning wordt de filosofie van de benadering globaal uiteengezet. De benadering staat de tuinder toe delen van de eerder genoemde transitieslag door het systeem te laten uitvoeren. Hierbij vertelt de tuinder het systeem wat voor doelstellingen het moet nastreven, waarna het systeem vervolgens de verzameling van instellingen bepaalt waarmee deze (zo goed mogelijk) worden gerealiseerd.

Op basis van de analyseresultaten worden in hoofdstuk zes het programma van eisen en het functioneel ontwerp van het systeemconcept uiteengezet. In het programma van eisen komen zowel allerhande systeemfuncties als de praktische randvoorwaarden die binnen dit probleemdomein van toepassing zijn aan de orde. In het functioneel ontwerp worden de systeemfuncties op een logische wijze ondergebracht in modules. Het systeemconcept bestaat uit twee deelsystemen met onderscheiden tijdconstanten. Het 'snelle' deelsysteem, met een tijdhorizon van ongeveer één uur, is verantwoordelijk voor de korte termijn regeling van het kasklimaat en de water- en nutriëntengift. Dit deelsysteem voert ook de metingen aan het kas-gewas systeem uit. Het 'langzame' inferentie- en interactiedeelsysteem is verantwoordelijk voor de interactie met de tuinder en leidt op basis van zijn invoer de randvoorwaarden (bijv. minimale en maximale nachttemperatuur) voor het regelmodule af. Kernmodule binnen het inferentie- en interactiedeelsysteem is de redeneermodule.

Deze module voert de conversie uit van variabelen op het gewas- en omgevingsniveau naar variabelen op het stuurniveau (d.w.z. de verzameling van variabelen die binnen het deelsysteem voor de (klimaat)regeling bekend zijn). Het inferentie- en interactiedeelsysteem bevat daarnaast modules voor allerlei praktische zaken, zoals: ondersteuning bij parameterijking, interactie met externe databanken, scenario-analyse, etc.. Tenslotte wordt in hoofdstuk zes op basis van de eigenschappen van het inferentieprobleem en de kenmerken van de beschikbare kennis, de keuze gemaakt voor 'constraint reasoning' als centrale inferentiemethode binnen de redeneermodule. De term 'constraint' laat zich in het Nederlands het best vertalen met de termen: randvoorwaarde of eis. Constraints beperken het waardebereik van variabelen.

In hoofdstuk zeven wordt vervolgens verder ingezoomd op constraint reasoning als inferentiemethode. Allereerst wordt de methodiek van constraint reasoning in algemene termen toegelicht. Vervolgens worden de voor dit proefschrift relevante delen uit het constraint reasoning vakgebied formeler en in meer detail toegelicht. De kern van dit hoofdstuk gaat in op de vraag op welke wijze de techniek van constraint reasoning toegesneden kan worden op de eigenschappen van het domein en de taken van de redeneermodule. Het gaat er om een passende representatie van de toestand en de kennis in het probleemdomein te vinden, alsmede de in- en output constraints op een adequate wijze af te beelden. Centraal in de hier gekozen modelleerwijze staan: *i*) het kunnen omgaan met onzekerheid en voorkeuren, *ii*) het gebruik van kwalitatieve en kwantitatieve kennis, en *iii*) de discrete en hiërarchische behandeling van de continu voortschrijdende tijd. Op basis van de gekozen tijdhiërarchie is de huidige of gewenste toestand van het kas-gewas systeem op het laagste niveau in hiërarchie bekend, of kan met behulp van constraintrelaties worden afgeleid. Individuele constraintrelaties kunnen worden gezien als (deel-)modellen uit het kennisdomein. Op basis van de toegepaste representatie is vervolgens een passend algoritme gekozen; het hier gekozen algoritme werkt op basis van interval domeinreductie (voor numerieke variabelen).

Tenslotte wordt aan het eind van hoofdstuk zeven beschreven welke delen van de theorie in een prototype zijn geïmplementeerd.

In hoofdstuk acht wordt met behulp van eenvoudige experimenten het gedrag van het prototype nader geanalyseerd. De experimenten demonstreren dat de gekozen interval benadering het werken met onzekerheid en preferenties en de integratie van kwalitatieve en kwantitatieve kennis eenvoudig mogelijk maakt. Daarnaast blijkt echter ook dat veel onzekerheid en brede preferentie-intervallen zorgen voor een problematisch groot aantal oplossingen.

In hoofdstuk negen worden de resultaten van het ontwerpdeel van dit proefschrift in samenhang bediscussieerd. Hier wordt met name ingegaan op de eigenschappen van de constraint reasoning als inferentiemethode binnen het totale concept. Het blijkt dat er ten aanzien van de praktische toepassing van constraint reasoning binnen het gekozen systeemconcept nog een aantal onzekerheden bestaan die alleen met behulp van grootschaligere experimenten opgehelderd kunnen worden. Het betreft hier: de gevolgen van de grootte van het inferentieprobleem, de representatie van bepaalde constraintrelaties en de algoritmekeuze. Tevens worden er strategieën aangedragen waarmee deze problemen kunnen worden aangepakt. Tenslotte wordt in dit hoofdstuk ingegaan op de interactie tussen de redeneermodule en de regelmodule uit het langzame resp. snelle deelsysteem. Uit deze discussie blijkt onder andere dat er verschillende strategieën toegepast kunnen worden om de door de redeneermodule gegenereerde oplossingen geschikt te maken voor de regelmodule.

In het laatste hoofdstuk wordt de onderzoeksdoelstelling nader bediscussieerd en wordt een scenario geschetst voor de verdere ontwikkeling van het systeemconcept. Ten aanzien van de onderzoeksdoelstellingen wordt betoogd dat het systeemconcept voldoet aan de eerder gestelde eisen, echter, de hoeveelheid werk die noodzakelijk is voor daadwerkelijke introductie in de praktijk, is aanzienlijk.

CURRICULUM VITAE

Pieter Johannes (Peter) Schotman werd geboren op 6 november 1964 in Dubbeldam (thans gemeente Dordrecht). In 1984 behaalde hij het VWO diploma aan het Christelijk Lyceum te Dordrecht. In dat zelfde jaar begon deze tuinderszoon aan de studie Tuinbouw aan de toenmalige Landbouwhogeschool te Wageningen. In 1990 slaagde hij voor het doctoraalexamen met de afstudeervakken Tuinbouwplantenteelt, Informatie en Bedrijfskunde. Na ruim een jaar in 's Lands leger gediend te hebben, trad hij eind 1991 voor een periode van vier jaar als 'Assistent In Opleiding' in dienst bij de Landbouwuniversiteit. Gedurende deze periode is de basis gelegd voor dit proefschrift.

Momenteel werkt hij voor de firma Bolesian BV, en begeleidt hij implemetatietrajecten van informatiesystemen ter ondersteuning van de productieplanning.