Physiology of Quality Change Modelling

Automated modelling of quality change of agricultural products

ato-dlo

SIKS Dissertation Series No. 99-1

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Graduate School for Information and Knowledge Systems, and in cooperation with the Agricultural Research Institute, ATO-DLO, Wageningen.

VRIJE UNIVERSITEIT

Physiology of Quality Change Modelling

Automated modelling of quality change of agricultural products

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit te Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
wiskunde en informatica
van de faculteit der exacte wetenschappen
op dinsdag 11 mei 1999 om 15.45 uur
in het hoofdgebouw van de universiteit,
De Boelelaan 1105

door

Mark Sloof

geboren te Schiedam

Promotor:	prof.dr. J. Treur
Copromotor:	dr.ir. M. Willems

Acknowledgements

This thesis describes the results of many years of research. During these years, a number of people contributed to this thesis in one way or another. Here, I would like to take the opportunity to thank them.

First of all, I am grateful to my promotor, Jan Treur, and my copromotor, Mark Willems. Although I was a relative outsider, both Jan Treur and Mark Willems always made time for supporting in doing research, and for accurate proof-reading of early and not so early drafts. I would like to thank Mark Willems for introducing me into the formalism of knowledge graphs, which form the basis for the process structure graph formalism described in this thesis.

The research was sponsored by the Agrotechnological Research Institute (ATO-DLO) in Wageningen, including a place to work. This made it possible to get to know the field of agricultural distribution chains, and of course the people working in this field. I am grateful to the people in the Agro and Industrial Production Chains division of ATO-DLO, who initiated this research, thereby in an early stage recognising the potential of quality change modelling of agricultural products. I would like to thank the division head Arjen Simons for being my supervisor, Jan Top for proof-reading earlier versions of the thesis, and all the other colleagues I worked with. My special thanks go to Pol Tijskens for the many long discussions we had about the concept of quality and quality change, and about modelling quality change of agricultural products. Although he was sceptical at first, very soon he became the greatest admirer of the research. Without his support, this thesis would not be what it is now.

Of course my thanks are extended to my reading committee (Hans Akkermans, Josse de Baerdemaeker, Frank van Harmelen, Cees Hoede, and Jan Top) for reading an earlier version.

I appreciated the conversations and discussions with my colleagues of the Department of Artificial Intelligence at (what is currently) the Faculty of Sciences at the Vrije Universiteit during the one day I worked each week in Amsterdam. Especially, Niek Wijngaards, Frederik Jan Jüngen, Pieter van Langen, and Pascal van Eck were kept from their work on those days.

And last but not least, I would like to thank everyone who gave me moral support during the whole period of writing the thesis, and helped me through the few periods of despair while finishing the thesis: friends, family, and of course my parents.

Contents

Chapter 1

Introduction

Quality of agricultural products is a very important aspect of our daily food. To persuade the consumer to buy a particular batch of products or to buy the products of a particular producer, producers have always tried to increase the external quality, such as the appearance of the product or the appearance of the package. In recent years, the internal quality attributes like taste, aroma, nutritive and constitutive compounds, have gained more and more attention.

As these internal properties are most of the time very difficult to measure or assess, legislations and regulations have been set up to provide safe guidelines for consumers and wholesale buyers. An example is a so-called tomato colour card to assess the quality of tomatoes. A colour card is a collection of photographs of a tomato at different stages of maturity and quality. By comparing the tomatoes in the batch with the photographs on the colour card, the stage of maturity of the offered produce can be assessed.

As these regulations tend to become more stringent, and consumers become more aware of internal quality, quality has become an important aspect of postharvest distribution of agricultural products. To predict the quality change during distribution requires that the relationships between the environmental conditions during distribution and the internal quality attributes are studied and modelled.

Both the external and the internal quality of a product are to a large extent the result of physiological processes occurring in the product. Recent years have shown a development towards explaining quality change in terms of these processes. Hence, quality change shall be modelled starting from the physiology of the product.

1.1 What is physiology ?

In *Webster's New Encyclopedic Dictionary*, physiology is defined as the 'branch of biology dealing with the processes, activities and phenomena incidental to and characteristic of life or of living matter'. Following this definition, the title of this thesis, *"Physiology of Quality Change Modelling"*, covers the main subjects under consideration in this thesis. In its literal meaning, the title refers to the fact that quality change of agricultural produce is most importantly the result of physiological processes occurring in the produce. As a metaphor, the title refers to the composite of activities that have to be conducted to model the aforementioned processes involved in quality change.

1.2 Quality and quality change models

After harvest, agricultural products remain living products. All kinds of physiological processes may continue or start after the moment of harvest. In principle, external nutrients are no longer available to the harvested product, so that the occurring physiological processes will use the resources that are internally available. As a consequence, during the distribution from the growers to the retailers and finally to the consumers, an agricultural product will change continuously, leading to changes in the quality of the product. Usually, quality change is a synonym for an unwanted loss of quality. However, sometimes quality changes are wanted changes, and may be induced e.g. to prepare an unripely harvested product for final use: many agricultural products are harvested at an immature state and allowed to maturate further during distribution. Examples are green bananas and hard melons. Another example is the chilling of flower bulbs to stimulate the onset of flowering. This treatment advances the ripeness state of the bulbs so that they will flower at a predetermined time.

To evaluate the effects of a distribution chain on the quality of the distributed agricultural product, *quality change models* are used. A quality change model is a simulation model for one product or a group of products, that describes the relation between the environmental conditions the product is subject to during distribution and the observed change in quality of the product. Because each activity in a distribution chain in some way affects the environmental conditions during (a part of) the distribution chain, quality change models can be used to evaluate all activities in a distribution chain. For example, repacking a product into a new package leads to a fluctuation in the gas composition and temperature in the environment of the product. A quality change model will take these changes in the environment as input to predict the effect of repacking actions on the quality of the product.

Quality change models can be used to evaluate alternative distribution chains with respect to quality change. A distribution chain starts from the moment of harvest. Hence, growth processes, that 'build up' the initial quality, are outside the scope of the constructed models and of this thesis. A distribution chain contains a number of 'quality assessment points', at which different definitions of 'good quality' are applied. For example, tomatoes purchased early in the distribution chain must be reddish, hence unripe, whereas tomatoes purchased by a supermarket must be red and firm. The quality definitions differ, but the physiological processes causing the quality change are the same.

An early example of a quality change model is the FLORES model [van Doorn and Tijskens, 1991], which is a generic model describing the loss in the *keeping quality* of FLOwers in RESponse to various and varying external conditions. The keeping quality is the number of days the flowers remain acceptable when kept at an optimal condition. In FLORES the keeping quality is called the *vase life* of the cut flowers. Loss of vase life occurs with the passing of time and if the environmental conditions are in some way suboptimal. The FLORES model consists of seven submodels. Each submodel describes the loss in vase life due to one effect. The total loss in vase life is the sum of the losses calculated by these submodels. The first submodel describes the basic decrease of vase life over a period of time and a certain temperature path. This effect occurs continuously, also at optimal conditions. The other submodels describe effects that further decrease vase life: dry storage, infection by fungi, bacteria growth, suboptimal temperatures, exposure to ethylene, and absence of flower preservative in the vase water. These effects occur if the flowers are in some way subject to suboptimal conditions.

The FLORES model has a number of desirable characteristics. First of all, the model is applicable to more than 20 different flower cultivars. Although the different flower cultivars show different responses to equal external conditions, the FLORES model uses one generic formulation for all flower cultivars, that is parametrised by cultivar specific sensitivity factors. For example, of the investigated flower cultivars only the *Carnation* cultivars show loss of vase life due to low temperatures. Hence, the sensitivity factor for low temperatures is zero for all flower cultivars in the FLORES model except the *Carnation* cultivars.

Furthermore, the total loss of vase life is described as a composition of seven causes for quality loss. These causes are described in separate submodels. The submodels exhibit minimal mutual interference and are dependent on a limited number of external conditions. In this way, the effects described in each submodel can be studied in isolation. Furthermore, minimal mutual interference and a well-defined and limited interface enhance the reusability of the submodels.

A disadvantage of the FLORES model is the combination of the physiological behaviour and the effects on quality. The functions in FLORES describe direct relationships between the environmental conditions and the loss of vase life. These functions do not explain the effects in terms of underlying physiological processes, which makes it impossible to reuse the functions in FLORES for other products. For example, not only flowers but also a lot of fruits are sensitive to ethylene exposure. However, as the ethylene function in FLORES is a direct relation between ethylene concentration and loss of vase life, the function cannot be used in other models to calculate the effect of ethylene exposure in other products.

The construction of quality change models is seen as a time consuming and difficult activity. The discussion of the FLORES model shows that this can attributed to an insufficient decomposition of the different disciplines of physiology and quality assignment. As a consequence, the models describe the effects of environmental conditions on the product using an implicit definition of 'good quality'. This combination of physiology and quality assignment has two consequences. Firstly, a model for one product cannot be reused for another product with a similar physiology, because the other product has a different definition of quality. Secondly, a model can be used only to simulate quality change according to one definition of quality. If different definitions of quality can be applied to the same product, then different models have to be developed for each definition of quality.

Product behaviour, and the resulting quality change, is the result of a combination of many processes with many mutual interactions. The FLORES model shows that linking the observed behaviour directly to changes in the environmental conditions, often yields simulation models with a narrow application area, and little explanatory value. Although the FLORES model has a good prediction performance and is very useful to study the remaining vase life of the flower cultivars included in the model, submodels for other effects on vase life cannot easily be included, nor can the submodels be reused for other products. A better approach is to analyse the product behaviour and decompose it into a set of well-understood physiological processes and their mutual interactions. In this way, a more explanatory simulation model is obtained, that may also be applicable to other situations than those used for the development of the model.

1.3 Research goals

The overall research goal addressed in this thesis is to develop a method to support a modeller in the construction of quantitative models for the simulation of quality change of agricultural products during postharvest distribution.

The research goal is refined into three subgoals:

1. to develop a definition of quality that can be used to assess the effects of alternative distribution chains on the quality change of an agricultural product.
2. to develop an architecture for a quality change model that separates the quality assignment from the physiological processes, so that the same model for the physiological processes can be used with different models for quality assignment.
3. to analyse the activities in modelling quality change of agricultural products as the result of interactions between generic physiological processes, and to develop a method to support in the construction of quality change models by reusing previously constructed models for the generic processes.

1.4 Outline of the thesis

The thesis can be divided into two parts. In the first part, a definition of quality and an architecture of a quality change model are developed. The second part describes our method for automated modelling of quality change. The method is called DESIMAL, which is an acronym for 'DEsign and Specification of Interacting MAthematicaL models'. It is also a misspelling of the word 'decimal' to refer to the function of constructing quantitative simulation models.

Chapter 2 presents a survey of several approaches to define quality of agricultural products. The main conclusion of this chapter is that the quality of a product has three dimensions: the user, the product itself, and the market situation.

In *Chapter 3* a conceptual model of quality and quality change of agricultural products is presented. A main characteristic is the distinction between the quality assignment to a product and the physiological behaviour of a product. Furthermore, an architecture of a quality change model is defined, in which the changes in assigned quality are described as a function of the user's quality notion and the behaviour of the product, and in which the behaviour of the product is described as a result of various and varying environmental conditions.

Chapter 4 presents a survey of approaches to automated modelling and to modelling support. In the context of this survey, the general architecture is given of the proposed Desimal approach to automated modelling of quality change of agricultural products. The tasks in this approach are elaborated in the next chapters.

Chapter 5 describes the first main task of the Desimal approach, called Qualitative Process Analysis. The goal of this modelling task is to decompose a quality change phenomenon into a number of generic and primitive physiological processes occurring in the agricultural product in which the quality change phenomenon is studied. This modelling task results in one or more qualitative Process Structure Graphs, each representing a possible process decomposition of the qualiy change phenomenon.

Chapter 6 describes the second main task of the Desimal approach, called Simulation Model Construction. This modelling task takes one Process Structure Graph as input, and selects suitable quantitative models for the processes into which the quality change phenomenon is decomposed. The result is one or more quantitative executable quality change models.

Chapter 7 describes the Desimal environment that has been developed to demonstrate the proposed modelling method. Furthermore, a number of modelling cases are elaborated.

Chapter 8 compares the Desimal method with the approaches to automated modeling reviewed in Chapter 4.

Part of the research presented in this thesis has been done together with others: The survey on approaches to quality (Chapter 2) and the development of the conceptual model for quality and quality change (Chapter 3) have been done together with L.M.M. Tijskens and E.C. Wilkinson. This research has been published in [Sloof *et al.*, 1996]. Chapters 4, 5 and 6 have been published in [Sloof, 1998]. Early descriptions of the DESIMAL method have been developed together with A.E. Simons, and presented in [Sloof and Simons, 1993; Sloof and Simons, 1994]. The representation scheme for physiological processes by means of knowledge graphs has been developed in cooperation with M. Willems, and presented in [Sloof and Willems, 1995]. Furthermore, [Sloof and Tijskens, 1995], [Top *et al.*, 1995b], and [Tijskens *et al.*, 1996] focus on the domain of physiology and quality change of agricultural products.

Chapter 2

Quality and quality management

> 'Quality ...you know what it is. But that's self-contradictory. But some things *are* better than others, that is, they have more quality. But when you try to say what quality is, apart from the things that have it, it all goes *poof*! There's nothing to talk about. But if you can't say what Quality is, how do you know what it is, or how do you know that it even exists? If no one knows what it is, then for all practical purposes it doesn't exist at all. But for all practical purposes it really *does* exist. What else are the grades based on? Why else would people pay fortunes for some things and throw others in the trash pile? Obviously some things are better than others ...but what's the 'betterness'? ...So round and round you go, spinning mental wheels and nowhere finding anyplace to get traction. What the hell is Quality? What is it?'
> **R. M. Pirsig — Zen and the Art of Motorcycle Maintenance**

2.1 Introduction

From the moment of harvest, agricultural products have a limited life because of loss of quality during the period between harvest and consumption, even when optimal conditions are used during distribution. This loss of quality may be large if products are not treated optimally.

Quality is becoming an increasingly important marketing factor both for producers and consumers. So, during distribution of agricultural products, the management of quality is very important. Because of this increasing importance, definitions for quality have been developed within various areas of research. These definitions of quality are reviewed in Section 2.2. In Section 2.3 techniques for the management of quality of agricultural products during postharvest distribution are reviewed.

2.2 Approaches to quality

Quality is a very elusive concept, that depends on many factors. In the first place, quality depends on the product itself. Quality also depends on the preferences of the user. The preferences may arise from the intended use of the product (e.g. ripe tomatoes for soup, hard tomatoes for salads), and from social-psychological factors such as the user's attitude towards the product. For example, one person may be status-conscious and prefer plum tomatoes from Italy, another may be environmentally aware and prefer organically grown tomatoes. A third aspect that may affect quality, is the market situation: the quality of a product depends on its price (a higher price is often taken to indicate higher quality) and on the availability of other, competing, products (a product of moderate quality will be assigned a higher quality when surrounded by products of poor quality, than when surrounded by products of high quality).

Several approaches to defining quality that reflect these different aspects have been described. Garvin [1984] identifies five approaches to defining quality:

- the *transcendent* approach, in which quality is viewed as an unanalysable property that an individual can only learn to recognise through experience;
- the *product-based* approach, in which one or more attributes of the product are used to compute a measure of quality: if these attributes are present in the desired intensities, the product has a high quality;
- the *user-based* approach, which places the consumer in the central position: quality of a product is based on how the consumer perceives it;
- the *manufacturing-based* approach, which uses technical specifications to objectify product quality: a product that conforms to the technical

specifications has a high quality;

- the *value-based* approach, in which the price-quality ratio is used to define the value of a product to the consumer: this approach assigns a high value to products with favourable price-quality ratios.

An extensive survey of literature on approaches to quality is given in [Steenkamp, 1989]. This author identifies four approaches to quality that stem from the areas of philosophy, production management, economics, and consumer research. In the following, these approaches are described in more detail.

2.2.1 Philosophy

The *metaphysical* [Steenkamp, 1989] or *transcendent* [Garvin, 1984] approach views quality as an unanalysable property that a user can only learn to recognise through experience. Because people acquire different experiences, their quality evaluations are bound to be different.

In *Webster's New Encyclopedic Dictionary*, metaphysics is defined as 'the part of philosophy that is concerned with the ultimate causes and the underlying nature of things'. Answers are sought for questions about 'the kind of things there are and their modes of being'. In metaphysics quality is considered something absolute and recognisable, but also something that cannot be defined precisely. However, quality must exist, as a world without quality would not function normally. For example, Aristotle made quality one of his 'categories' with which a substance is specified. According to Aristotle, quality cannot exist without the substance. The citation at the beginning of this chapter, from Pirsig's book '*Zen and the art of motor cycle maintenance*', also expresses that quality must exist, even though it cannot be defined without referring to some substance.

Descartes made an important distinction between 'primary qualities' and 'secondary qualities'. Primary qualities are inseparable from the object. Secondary qualities are properties of an object as they are perceived by humans. These qualities are subjective and derived from primary qualities. For example, the colour of a tomato is a secondary quality that is the human perception of the way in which the cell structure of the tomato (a primary quality) reflects and refracts light.

2.2.2 Production management

The *production management* approach is concerned with maintaining quality during production, and uses technical specifications to objectify product quality. Thus, a product that conforms to the technical specifications has a high quality. Production starts with product design, includes manufacture and distribution, and extends to maintenance and after sales services. For each stage in the production process, in [Juran *et al.*, 1974] specific quality criteria are defined to monitor and control that production stage:

- *quality of design* is the degree of excellence the product must possess. A product has a high quality of design if already during design of the product quality plays an important role, e.g. by designing for a reliable, easily maintainable product that will fulfil the needs of the consumers for whom the product is intended. A high quality of design can be achieved by (1) identifying the quality requirements of consumers, (2) developing a product concept meeting these requirements, and (3) translating the product concept into technical specifications. The quality of design defines an upper limit to the objective aspects of quality of the individual products. The meaning of this parameter is that a poorly designed product cannot get a higher quality by improving the production process or by improving the after sales service, but only by redesigning the product or a part of it.
- *quality of production* refers to organising the production process so that the products meet the technical specifications. Quality of production can be monitored either as a final inspection at the end of the production process or by controlling the production process and making immediate adjustments if necessary.
- *continuity of service* is concerned with the reliability and maintainability of the product. Thus, products that operate satisfactory and can be easily repaired when necessary, have a higher quality than products that often need repair.
- *consumer service after sale* includes speed, competence, and integrity of the after sales service. This is more a marketing aspect of quality: the quality of the product is increased by better instructions for the sales department.

Although the production management approach was developed for nonperishable products, the concepts relating to the design and the production stages can be applied to agricultural products as well. Breeding new cultivars with properties such as a better resistance to certain diseases or a better taste, can be regarded as improving the quality of design of an agricultural product. For example, a new apple cultivar may be developed that is particularly sweet (a desirable quality attribute). Ripening, occurring during the distribution, can make sour apples sweeter, but not to the same degree as apples that are sweet in nature.

The production of agricultural products consists of a growth phase and a distribution phase. Examples of maintaining quality during production are control of the growth conditions, and use of packaging throughout a complete distribution chain.

The last two parameters, continuity of service and consumer service after sale, refer to maintenance and repair of the product and how this is handled by the producer. They are therefore only applicable for durable products such as cars and computers. In the context of quality management of perishable products these two parameters are not relevant.

2.2.3 Economics

Economic theories of producer and consumer behaviour in markets containing products of differing quality use a value-based and product-based definition of quality. In this case, quality is usually a composite of product characteristics.

Theories about producer behaviour describe how producers use quality to maximise their profits by differentiating their products from competing products. Differentiation can be achieved by (1) changing the value of a quantitative characteristic, for example by increasing the amount of vitamins in a food product (vertical quality variation); (2) by making the product more appealing to a specific group of consumers, for example by harvesting fruit at different time points, so that consumers can choose between ripe and unripe fruit (horizontal quality variation); or (3) by introducing a new quality attribute that eventually may make existing quality grades obsolete, for example using organic growing methods instead of using mineral fertilisers (innovative quality variation).

Economic theories about consumer behaviour on markets with products with differing quality assume that consumers try to buy those goods that have the highest quality. In [Lancaster, 1971], quality is defined as 'those objectively measurable, technical properties of goods that are relevant to consumer choice'. Different consumers may perceive these properties differently. In Lancaster's model the differences in perception are captured in individual preference functions.

Many economic theories assume that consumers are completely informed about the price and quality of the products available on the market. This assumption is, however, unrealistic. Most of the time, consumers are imperfectly informed and, therefore, use various strategies to evaluate the quality of available alternatives. Depending on how the quality of a product is determined, three types of strategies can be distinguished. The first is to *search* for a product with the highest quality by inspection of available products prior to purchase. An example is comparing available wines by using descriptions of the bouquets. The second strategy that can be used to evaluate the quality of product alternatives is by *experience*: by trying different alternatives and selecting the one that provides the largest benefit. An example is to buy and taste different wines, until a wine with the most favourable bouquet is found. Some attributes cannot be evaluated from actual experience with the product. In case of these so-called *credence* attributes, consumers must rely on information from external sources. An example of a credence characteristic is the percentage alcohol in the wine.

2.2.4 Consumer research

The *user-based* or *perceived-quality* approach puts the user in the central position. In this approach, quality is considered to be subjective: it depends on the perceptions, needs and goals of the individual user. The terms '*perceived* quality' and 'fitness for use' [Juran *et al.*, 1974] emphasise this.

Several definitions for perceived quality have been proposed. [Kramer and Twigg, 1983] define quality as 'the composite of those characteristics that differentiate individual units of a product, and have significance in determining the degree of acceptability of that unit by the buyer'.

In [Böckenhoff and Hamm, 1983] quality is defined as 'the composite of all product attributes irrespective of whether these attributes are in reality existent in the product and objectively measurable, and whether consumers are correct in their evaluations'. This definition expresses that the consumer may attribute certain properties to a product that in reality do not exist, which is an important difference with the product-based approaches, using only measurable product properties.

Steenkamp [1989] gives the following extensive definition of quality:

> [Quality can be defined as] 'an idiosyncratic value judgement with respect to the fitness for consumption of the product which is based upon the conscious and/or unconscious processing of appropriate and available intrinsic and extrinsic quality cues in relation to relevant experience and credence quality attributes, and formed within the context of prior experience, perceived quality risk, quality-consciousness, usage goals, and other personal and situational variables' [Steenkamp, 1989, p. 107].

In this definition a distinction is made between quality cues and quality attributes. *Quality cues* are those product-related characteristics that are ascertained prior to consumption. Quality cues are similar to the search attributes of [Lancaster, 1971], and can be either intrinsic or extrinsic [Olson and Jacoby, 1972]. Intrinsic quality cues are part of the product, and cannot be changed without also changing the nature of the product. Examples are firmness and colour. Extrinsic quality cues are related to, but not part of, the product. Examples are brand name and price. *Quality attributes* are observable only during or after consumption. Two types of quality attributes are distinguished: *experience attributes* and *credence attributes*, which have the same meaning as in Lancaster's economic model of consumer behaviour. The former can be ascertained on the basis of actual experience with the product, the latter cannot be ascertained even after using the product for a long time.

Based on prior experiences and on evaluations of credence attributes, a consumer decides whether or not to buy a certain product again. This demonstrates that consumers are imperfectly informed about quality of products available at the market: preferably, a consumer wants to know the values of the quality attributes before buying a product. As this is impossible, a consumer selects and evaluates those quality cues, that are believed to relate to quality attributes. These beliefs can be formed either through direct observation of experience quality attributes, or through inferencing ('if the colour of an apple is red then the apple is ripe enough'), or by accepting information from external sources such as consumer magazines. Direct observation of experience quality attributes can be done by trying the product before purchase, e.g. by tasting a product sample. In this case values of the quality attributes can be determined directly and quality cues are not needed.

Which quality cues are selected depends on the *quality information content* of the available cues. The cues that are thought to provide the most information about related quality attributes are selected. [Olson and Jacoby, 1972] define three dimensions for the information content of a cue:

- The *predictive value* of a cue is the extent to which the consumer perceives or believes that the cue is related to or is indicative of product quality. The more a cue is seen as related to product quality, the larger its quality information content will be.
- The *confidence value* is the degree to which a consumer is confident in his ability to accurately perceive and judge the cue. The quality information content of a cue is larger if its confidence value is large.
- The third dimension is *intrinsic-extrinsicness* of a quality cue. Intrinsic cues cannot be changed or manipulated without also changing the physical characteristics of the product itself. Extrinsic cues are related to the product but are not part of it, examples are brand name and price. Therefore, intrinsic cues have a larger quality information content than extrinsic cues.

These dimensions are related as follows: intrinsic cues with a high predictive value and a high confidence value will be used as much as possible. These cues are real product properties that according to the consumer are strongly connected with product quality, and that the consumer judges himself capable of to evaluate correctly. Consumers will avoid using cues with a low predictive value or a low confidence value, either because they perceive the cues as not indicative for product quality, or because they are afraid to misjudge such cues. If the available intrinsic quality cues have low predictive and low confidence values, a consumer will resort to extrinsic cues to get information about the quality of a product.

2.3 Integrated Quality Management

Defining quality and modelling quality change is only part of the problem. Management of quality is equally important, because products do not automatically have the desired quality, and perishable products, in particular, may even deteriorate during the production process. Integrated Quality Management (IQM) is an important goal in many economic sectors, including the agricultural sector.

The production process of agricultural products is divided into two phases: a growth phase and a distribution phase. In both phases quality management is needed to achieve a desired quality level. For example, quality of greenhouse produce is managed by controlling the greenhouse climate and the fertilisation. In the distribution phase of the production process of agricultural produce, a form of quality management is applied to as much as possible avoid loss of the quality acquired in the growth phase. However, as some products are harvested before they are ripe, quality management during distribution can also be aimed at improving the harvest quality of such products.

An early methodology for the management of quality of horticultural products during distribution is presented in [Schoorl and Holt, 1985]. The methodology consists of three 'subsystems', quality prediction, market price prediction, and management decision making:

- *Quality prediction* In the first subsystem a product quality is predicted from the input parameters product, packaging, storage, transport, handling and transit time. Three forms of postharvest deterioration are considered: mechanical damage, physiological deterioration, and pathological decay.

  Mechanical damage may occur during transport or handling of the products. Bumpy roads or drops during handling are mentioned as causes of the generation of mechanical energy. If this energy cannot be absorbed sufficiently by the package, the product inside the package may be damaged. Mechanical damage is modelled by calculating the energy input to the products and using damage-energy relationships. Schoorl and Holt mention two ways to decrease quality loss due to mechanical damage: (1) using packages with better energy absorption properties, and (2) distributing the products at an earlier state of ripeness, in which most products are firmer and therefore less sensitive to mechanical damage.

  Physiological deterioration is mainly caused by suboptimal environmental conditions during distribution. Schoorl and Holt mention injury, senescence and water loss as important forms of physiological deterioration. Injury can occur in various forms depending on different unfavourable environmental conditions, examples are chilling injury caused by too low temperatures, alcoholic breakdown due to anaerobic respiration, and injury due to exposure to ethylene.

  Pathological decay is caused by growth of bacteria, yeasts or moulds on places where the product is injured, possibly because of mechanical damage. Because pathological decay is a growth process, its effects may remain hidden for some time. Therefore it may be difficult to find out what may have caused the initial infection. Schoorl and Holt mention that the growth process of most postharvest diseases can be described in terms of a lag phase, an exponential phase and a saturation phase, that is, the growth process is a logistic function of time.

- *Market price prediction* The second 'subsystem' takes as input the product quality as predicted in the first subsystem of the methodology to determine the market price of the product. First a quality index is calculated that depends on the predicted product quality and the average quality of comparable products available in the market. This quality index indicates the superiority of the product with respect to the other products. Together with supply-demand and quality-price relations the quality index leads to a market price.

  The quality-price relation of a product depends on the reputation for quality of that product and on the perceived quality at point-of-sale. Schoorl and Holt define reputation as something that is 'established by quality attributes which are taste- or time-dependent; that is, ones that can only be assessed after the produce has left the markets, either by the retailers or the consumer — for example, shelf-life or texture'. Quality at point-of-sale is assessed using attributes such as bruising, water loss, size, and shape. Also pack presentation and salesmanship at the market floor are mentioned as attributes for point-of-sale quality of a product. These latter attributes are, however, less important than the product-related attributes.

- *Management decision making* The third and last subsystem is concerned with the economic evaluation of alternative distribution systems, each leading to different price cost ratios. It takes the market price as predicted in the previous subsystem as input. The aim is to maximise profit during distribution, which does not necessary imply delivering the best possible quality. As a high quality level may only be reached with high costs which may reduce the profit, this subsystem involves the trade-off between improving the product quality, and the costs needed to reach the higher quality.

The methodology gives a detailed description of the activities that are needed to prevent quality loss during the distribution of agricultural products, and provides a way to construct a model for the quality change of agricultural products during postharvest distribution. Because the activities of each 'subsystem' are explicitly mentioned, the methodology seems to be focused on a certain class of distribution chains. From this methodology it can be derived which aspects of the distribution chain are important for quality management. Using this methodology for an actual distribution chain, these aspects can be identified, and the effects of the actual activities in the distribution chain on these aspects can be modelled.

The prediction of the product quality in this methodology is based on the activities performed with the products rather than on basic physiological processes occurring in the product. This may pose problems when applying the methodology to products that are treated differently, or when the activities change. For example, only the mechanical properties of the packages are used, while for Modified Air packages the permeability properties to various gases is more important for the quality of the packaged products. This way of modelling quality change — i.e. starting from the activities rather than from the physiological processes — was also followed in the FLORES model for the prediction of the keeping quality of cut flowers [van Doorn and Tijskens, 1991].

The approach taken in the market price prediction corresponds to the model of perceived quality described in [Steenkamp, 1989]. The perceived quality of a product depends on previous experiences with the product leading to a certain reputation, and on the conditions of the product at point-of-sale as perceived by evaluating available attributes. The quality attributes that establish reputation for quality in the methodology of Schoorl and Holt are the experience attributes in Steenkamp's model, and the attributes used to evaluate point-of-sale quality are called quality cues in the model of Steenkamp. Steenkamp additionally takes into account the effects of information from external sources (credence attributes) on perceived quality, which are not used in the methodology of Schoorl and Holt.

2.4 Conclusions

Although the theories to defining quality stem from different areas of research, in all approaches, the quality of a product is a combination of three factors: the product, the user, and the market situation.

Product-related factors In all theories described above, quality of a product is defined as a composition of different types of attributes of the product. In product-based approaches the attributes are measurable, while in the user-based approaches also attributes may be used that in reality are not existent in the product but that, according to the consumer, are important for the quality of the product. The distinction between primary and secondary qualities, made by the philosopher Descartes, is important as it demonstrates that the outlook of a product is a secondary quality, that in fact depends on properties of the product which may not be visible or measurable (primary qualities). The intrinsicness-extrinsicness dimension [Olson and Jacoby, 1972] shows that only a small number of the attributes that a consumer uses to evaluate the quality of a product really belong to that product. Varying the other, extrinsic, attributes does not change the product itself. Such extrinsic quality attributes can be manipulated to make a product more appealing to a consumer group (horizontal quality variation).

User-related factors An important contribution from the philosophical approach to the concept of quality is that an individual can only learn to recognise quality through experience. Because people have different experiences, also their quality evaluations are different. This is also reflected in the individual preference functions in the economic model of [Lancaster, 1971]. The distinction between product properties and the perception of these properties by consumers is an important aspect of the conceptual model of quality change to be presented in Chapter 3. The model of Lancaster may be used to describe the relations between quality aspects and product properties.

Market-related factors Quality of a product depends also on the market situation. Here, the extrinsic quality attributes are important, as they differentiate a product from other competing products, without leading to a difference in the 'intrinsic quality' of the product. These extrinsic attributes are used by a consumer when deciding on whether or not to accept and buy the product. A consumer then asks himself whether he is willing to pay the price asked for a product taking into account the brand and the perceived quality of the product.

Chapter 3

A model for quality and quality change

3.1 Introduction

The previous chapter discussed several definitions to quality showing that quality depends on three factors: the product itself, the user of the product, and the market situation at the moment of purchasing the product.

In this chapter we will show that an explicit decomposition of the effects of these factors on the quality of a product will lead to a distinction between the assigned quality and the acceptability of a product. Assigned quality is the quality notion a consumer has of a product, and results from evaluating that product with respect to the consumer's specific quality criteria. Acceptability defines whether the consumer in a particular situation is willing to buy a particular product, and is the result of relating the product's assigned quality to other products and to extrinsic factors such as the price.

For monitoring and controlling changes in assigned quality during postharvest distribution, separate models can be developed: (1) for the quality assignment by a defined group of users, (2) for the physiological behaviour of the product, and (3) for the environment of the product. These three models can be composed into a quality change model, describing the changes in quality of a product in various and varying environmental conditions.

The outline of this chapter is as follows: In Section 3.2 the conceptual model of quality and quality change of agricultural products is defined. In Section 3.3 a quality change model is defined as the composition of separate models for the quality assignment, for the product behaviour, and for the product environment.

3.2 A conceptual model of quality and quality change

In the approaches described in the previous chapter, the quality of a product depends on both intrinsic and extrinsic product properties. The intrinsic product properties define the state of the product, which is evaluated with respect to quality criteria imposed by a producer (product management approach) or by a consumer (consumer research approach). Extrinsic product properties, such as the price and the quality-price ratios of the product and of other products, are used as additional information in the decision whether or not to purchase the product.

The distinction in the use of intrinsic and extrinsic product properties can be extended into a distinction between the *assigned quality* of a product and the *acceptability* of a product. This is illustrated in Figure 3.1.

Assigned quality is the result of an evaluation of a product only with respect to the intrinsic product properties. Assigned quality specifies the suitability of the individual product to the needs and goals of a user, without referring to extrinsic properties of the product, or to other products. The needs and goals of the user are reflected in the criteria that the user imposes on the intrinsic product properties, when assigning quality to the product. The grey bar in Figure 3.1 between 'intrinsic product properties' and 'assigned quality' indicates that the user plays an active role in the quality assignment: by defining criteria for the intrinsic product properties the user determines what quality is assigned to a product. As an example, a different criterion will be applied to the firmness of a tomato according to whether it will be used in a soup or in a salad: ripe tomatoes are more suitable for soup, whereas hard tomatoes are more appropriate for salads. Therefore, ripe tomatoes will have a high assigned quality if the user wants to make tomato soup, but the same tomatoes will have a low assigned quality, if the user wants to use them in a salad.

The combination of the assigned quality, the extrinsic product properties, and

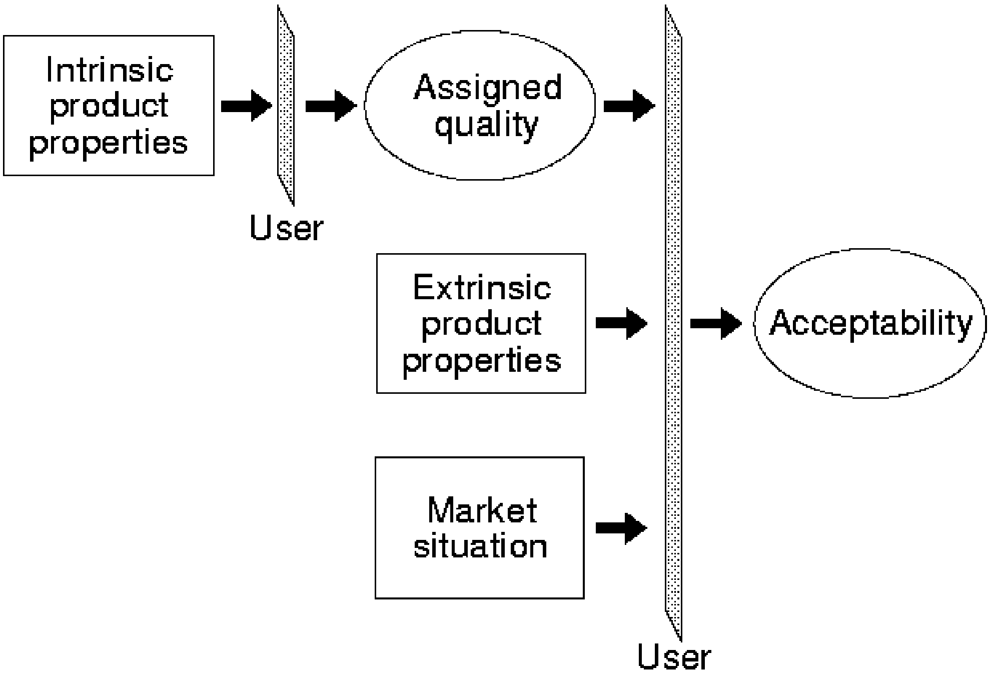


Figure 3.1: A user evaluates intrinsic product properties to assign quality to a product. By also taking into account extrinsic product properties and the market situation, the user determines the acceptability of the product.

the market situation yields the *acceptability* of a product: an assessment of the product in relation to its price and to other products. Independent of the assigned quality of the product, the acceptability will decrease or increase if other products are assigned a better or worse quality, respectively. The acceptability of a product corresponds to 'affordable excellence' [Garvin, 1984], that is, to its quality in terms of costs and price. This value-based approach to quality is often difficult to apply, as it combines a measure of excellence (quality) with a measure of value (price). Here again, the user plays an active role, as is indicated by the second grey bar in Figure 3.1 in front of the 'acceptability' ellipse. It is the user who evaluates the other products and the price-quality ratios. Thus, the evaluation is affected by socio-psychological factors, such as quality-awareness, status-awareness, etc. of the user.

From this perspective, the approaches reviewed in Chapter 2 describe strategies used by consumers in deciding whether to accept a product (the economic theories about consumer behaviour and the consumer research approach), and strategies used by producers to increase the acceptability of their products (the economic theories about producer behaviour and the production management approach). The concept of perceived quality used in consumer research differs from assigned quality, in that the perceived quality depends also on extrinsic product properties and on the market situation, whereas the assigned quality depends solely on intrinsic product properties.

Assigned quality may change because of changes in intrinsic product properties, or because of changes in the criteria imposed on these product properties. This distinction between changes in product behaviour and in quality criteria can also be found in models for keeping quality [Tijskens, 1995; Tijskens and Polderdijk, 1996]. Keeping quality is defined as 'the time a product remains acceptable under whatever circumstances and using whatever acceptance limits'. Like assigned quality and perceived quality, keeping quality is a combination of the product behaviour and of the (possibly changing) quality criteria. However, keeping quality differs from assigned quality and perceived quality, in that the latter two represent assessments of a product at a certain point in time, whereas keeping quality represents the period of time that all quality attributes of the product comply with the quality criteria.

Changes in the intrinsic properties of agricultural products may be caused by conditions in the environment to which the products are subjected during postharvest storage and distribution. The environmental conditions themselves can be affected by the product, particularly in the case of packaged products; for example, respiring fruits give off carbon dioxide, thereby changing the environmental conditions. In such a case, a strong bi-directional interaction exists between the product and its environment. For non-packaged products, only a uni-directional interaction is important, because the influence of such a product on its environment is negligible.

From this line of reasoning, the changes in the assigned quality of agricultural products can be decomposed into three quality-determining factors: the assignment of quality to a product by the user, the changes in the intrinsic product properties, and the interaction between the product and its environment.

3.2.1 Quality assignment by the user

Users select certain quality attributes and impose criteria on these attributes to assign quality to a product (see Section 2.2). The quality attributes selected by a user, and the criteria imposed on them, form the *quality notion* of the user with respect to a certain product. Although each user may in principle have a different notion of quality, groups of users can be identified that use the same quality attributes in their evaluations, and impose more or less equal criteria on these quality attributes. Examples of such groups are households with growing children, vegatarians, households with a double income, etc. We call such groups *homogeneous* with respect to assignment of quality to a certain product. In modelling quality change, quality assignment is defined with respect to such homogeneous groups of users rather than with respect to an individual user.

The assignment of quality to a product is a process that, in several steps, transforms the many intrinsic properties of a product into one (subjective) uni-dimensional measure of quality. To arrive at an assignment of quality, a user perceives and evaluates a number of intrinsic product properties, and then carries out an appreciation of these evaluations. The three steps of quality assignment are depicted in Figure 3.2. In the rest of this section these three steps are explained in more detail.

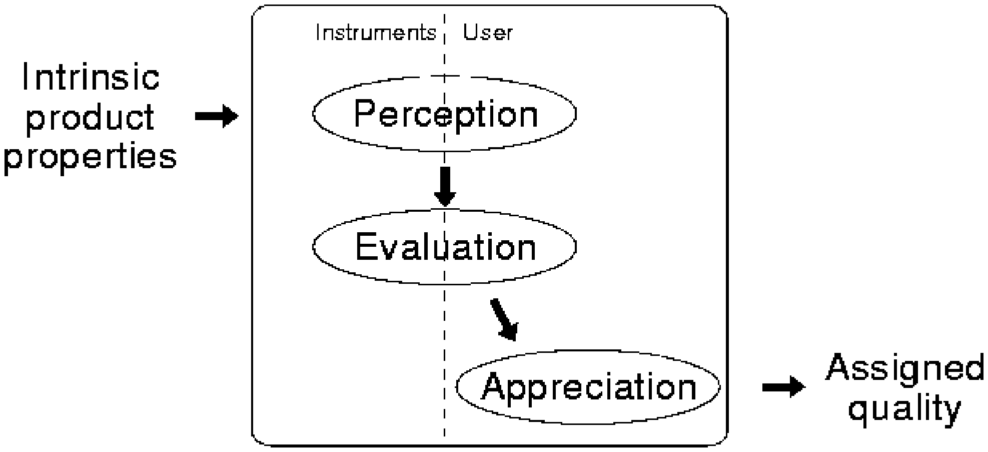


Figure 3.2: Steps in quality assignment to a product by a user. For the perception and evaluation steps both instruments and human senses may be used, the appreciation is entirely done in the mind of the user.

Perception

The first step in quality assignment is the perception of the intrinsic product properties. Properties of perishable products can be perceived either by instruments (e.g. firmness can be measured by a penetrometer or an Instron, colour by a colour meter) or by human senses (e.g. firmness of a tomato can be assessed by pressing the tomato between your fingers). Some properties, such as the vitamin C content, can only be measured using instruments. These are the so-called hidden attributes [Kramer and Twigg, 1983]. Other properties can (to date) best be assessed by human senses (e.g. flavour). Through perception, the intrinsic properties of a product are converted into quality attributes. A single quality attribute can be based on several product properties. A good example is colour, which in most cases is the perception of the combined concentrations of several colour components inside the product.

Sensory perception is complex. Even mealiness of apples, which is an apparently straightforward quality attribute, does not show a one-to-one relationship with the amount of cell juice, but depends also on how the apple tissue fractures when bitten. Mealiness is enhanced by fracture along cell walls, thereby preventing the perception of the juice and the sugars present in the intact cell. A user would therefore experience a mealy apple as dry, although the apple may contain almost the same amount of juice as a crisp apple.

Evaluation

In the second step, the perceived quality attributes are evaluated to determine their intensities or values. Evaluation can also be conducted both by instruments and by human senses. As perception and evaluation are strongly connected, perception and evaluation of a quality attribute are usually performed with the same 'equipment', that is, using instruments or human senses.

The relation between a stimulus intensity and the corresponding sensation experienced by the human senses is not a simple linear one. It generally flattens at high intensities owing to saturation of the human senses, whereas intensities below a certain threshold intensity that is specific to the user, will not be perceived at all.

Another characteristic of using human senses instead of instruments to evaluate quality attributes is a possible shift in perceived intensity following the evaluation of several products. Thus, a quality attribute may be evaluated differently if it is assessed after a batch of products with low intensities for that attribute than after a batch of products with high intensities.

Appreciation

Once the quality attributes are perceived and evaluated, they can be converted into appreciations. In many cases, the relation between evaluation and appreciation of a quality attribute shows a strong optimum: following the first increase in liking with increasing intensity, the curve flattens in a region of no preference, which is followed by a more or less steep decline in liking with increasing intensity. As an example, a very weak salt solution is not very agreeable, neither is a very strong salt solution.

Finally, the appreciations of the individual quality attributes are combined into a uni-dimensional quality measure. In this step, relative weights are assigned to the individual quality attributes and to combinations of quality attributes. These weights reflect the influence of socio-psychological factors, including personal preferences, trends, tradition, and status symbols, on the assignment of quality to a product. The socio-psychological factors determine the attributes to be used, and the order of importance of these attributes.

3.2.2 Describing a product state

During the quality assignment, users evaluate and appreciate quality attributes that are perceptions of properties of the product. Product properties are described using three dimensions: intrinsic or extrinsic, variable or fixed, and whether or not the property can effectively be changed.

Firstly, it must be determined whether a product property is intrinsic or extrinsic. The definition of this dimension corresponds with the distinction between intrinsic quality cues and extrinsic quality cues used in the consumer research approach (see Section 2.2.4). An *intrinsic product property* is inseparably related to a product. So, changes in intrinsic product properties will always lead to changes in the product and may therefore affect the product quality. An *extrinsic product property* can change without automatically changing the product. For example, intrinsic properties of mushrooms are the species, the growing origin and conditions, the amount of water in the mushrooms, the firmness and the colour, whereas the price, the appearance of the package, and the shop where the mushrooms are bought, are extrinsic properties.

The second dimension for describing a product property is whether or not the property changes during the normal lifetime of the product. Properties that change during the lifetime of the product are called *variable product properties*; properties that are constant are called *fixed product properties*. Of the mushroom properties mentioned above, the growing origin and the species are fixed product properties, whereas the amount of water, the firmness and the colour are variable product properties.

The third distinction is whether the value of the product property can be effectively controlled or manipulated. This issue is more apt for the operational and strategic planning of the distribution of (agricultural) products than for modelling or understanding the postharvest behaviour of agricultural products. Operational planning concerns the performance of activities during distribution and, therefore, only concerns those variable product properties that can be effectively manipulated. Strategic planning, however, involves the (re)design of distribution chains; in this case properties that are fixed during the postharvest life of the products, such as harvest time, as well as variable product properties may be manipulated.

For the purpose of quality change modelling, only intrinsic product properties are relevant. Of these, the values of the variable product properties at any point in the lifetime of the product determine the *product state*. The product behaviour is a series of such product states at successive time points. Each product state has an assigned quality associated with it, which is determined by the user through the perception, evaluation, and appreciation of the product, as described in Section 3.2.1. Hence, the quality change of the product can easily be determined, given the time series of product states.

3.2.3 Behaviour of a product

During the normal lifetime of a product, the variable intrinsic product properties change as a result of processes occurring in the product. Examples of such processes include the (further) ripening of fruit, and the opening of broccoli buds. Many processes are complex systems of chemical reactions (respiration, colour development), whereas other processes have a physical nature (osmosis, diffusion). Yet other processes have both chemical and physical aspects, such as the complex process affecting the firmness of a product. Firmness may be described as a combination of turgor pressure, which is a physical quantity, and of the concentrations of various chemical compounds like pectins, which are affected by chemical reactions.

Each process causes changes in one or more variable product properties. The action of a process may be affected by external factors, such as ambient temperature, as well as by other product properties, both fixed and variable. Thus, a variable product property that is affected by one process may influence the action of another process in the product, that acts on another variable product property. These patterns of interactions between processes result in the observed complex physiological behaviour of agricultural products.

During the lifetime of a product, processes may be activated or inactivated. For example, during the distribution of vegetables in Modified Air packaging (MAP), the respiration, and consequently the rate of deterioration, gradually decreases because of the low oxygen concentration, but increases again quite suddenly when the package is removed. As another example, during the distribution of cut flowers, water uptake through the stem may become blocked because of growth of bacteria in the stem, or because of drying out due to shortage of water. Subsequent re-cutting of the flower stem may remove the bacteria and may re-enable water transport [van Doorn, 1993].

Apart from such discrete events as removing an MA package or flower stem re-cutting, a process may also become activated or inactivated as a result of a continuous change in the product. For example, many fruit in the pre-mature stage will ripen slowly until they reach the climacteric stage (a hyperactive state in many fruit just before ripening). On reaching this stage , the rates of ripening processes will increase, so that the effects of these processes become important, and the processes become activated. Another example is an enzymatic process that gradually becomes inactive due to denaturation of the enzyme, caused by blanching.

3.2.4 The environment of a product

As stated above, many processes in agricultural products are affected by conditions in the environment immediately surrounding the product. The product environment is represented by external factors, of which the most important are temperature, relative humidity, and the concentrations of oxygen, carbon dioxide and ethylene. Hence, to describe the product behaviour and the resulting quality change during postharvest storage and distribution, the changes in the product environment have to be modelled. Changes in the external factors are caused by physical processes occurring in the product environment, such as diffusion of gases through a package, and are caused by activities during distribution, such as cooling.

The external factors may also be affected by processes occurring in the product itself: respiration affects the oxygen and carbon dioxide concentrations, evaporation increases the relative humidity, and heat production changes the temperature in the environment. The effect of the product on the environment will become relevant when the product is contained in a relatively small closed space, for example inside a layer of packaging. Under such conditions, the environment may become unfavourable for minimising the loss of product quality; for example, evaporation of a product wrapped in a foil may cause the relative humidity inside the foil to rise, causing fungal infections [van der Sman *et al.*, 1996]. On the other hand, in the case of MAP, the package material is designed to exploit the processes occurring in the product to bring about an environment that is favourable for minimising quality change [Kader *et al.*, 1989].

3.2.5 Discussion

The conceptual model presented in this section aims at the description of quality change during postharvest distribution. As a consequence, the effect of economic factors on the acceptability of a product is largely neglected. The structure of a quality change model, defined in the next section, only takes into account the quality assignment by the user, and the interaction of the product with its environment.

A quality change model must combine the results of two areas of quality research. The first is consumer research, in which the differences in quality assignment between groups of individuals defined by their socio-economic characteristics (personal, national, regional and social preferences) are investigated. In this area of research, the economic factors and the product are assumed to be constant. The second is product research, in which the influence of product properties on quality assignment by the buyer, and the relations between product properties and quality attributes are investigated. In this area of research, the economic factors and the socio-psychological factors of the buyers are assumed to be constant.

A third area of quality research is market research. Assuming that the socio-psychological factors and the product quality are constant, market research aims at describing the mechanisms of the market, the effects of advertising, and relevance of availability on the sales of a product. This area of research is important when studying the changes in the acceptability of a product.

3.3 Definition of a quality change model

To model the changes in the quality assigned to a product that is subjected to particular environmental conditions, three separate models are needed: to describe quality assignment, to describe the product behaviour, and to describe the product environment. These three models and their interactions are depicted in Figure 3.3. The models will be described from right to left.

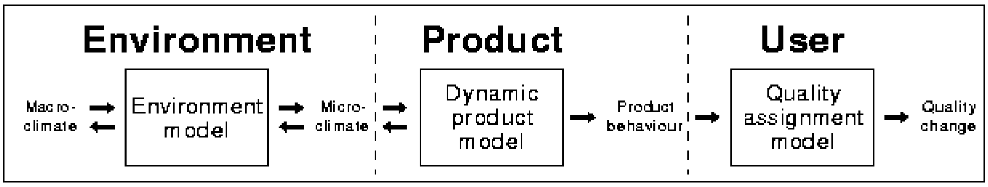


Figure 3.3: A quality change model is a composition of three separate submodels for the three causes for changes in assigned quality: the quality assignment by the user, the physiological behaviour of the product in a certain environment, and the changes in this environment.

3.3.1 The quality assignment model

A *Quality Assignment Model* (QAM) describes how a homogeneous group of users assigns quality to a given product. The QAM specifies the product properties that are relevant for that particular group of users. For each relevant product property, the QAM specifies the relationship between the product property and its appreciation by the group of users. Furthermore, the QAM contains a *quality function* that combines the appreciations of the individual product properties into one uni-dimensional measure of quality. This quality function can be compared with the preference function in the model of [Lancaster, 1971].

Wilkinson and Polderdijk developed a Quality Assignment Model for the assignment of quality to tulip bulbs by various user groups in the Dutch tulip bulb chain [Wilkinson and Polderdijk, 1995; Wilkinson and Polderdijk, 1996]. In this model, the quality function is a summation of individual quality attributes, of squares of quality attributes, and of products of pairs of quality attributes. The summations are weighted by weight factors that are based on the results of questionnaires sent to members of the links in the distribution chain. The first summation (of the individual quality attributes) represents the quality attributes with linearly increasing or decreasing appreciations, such as bulb damage and bulb disease (less damage and less disease always give a higher quality). The second summation (of the squares of individual quality attributes) represents the quality attributes with an optimum appreciation. The last summation (of the products of individual pairs of quality attributes) represents appreciations of combinations of quality attributes. For bulb quality, such an interaction exists between bulb damage and bulb disease: if bulbs are damaged less weight is given to the presence or absence of disease. This reflects the knowledge of the users that a damaged bulb is more susceptible to disease whether or not any disease is visible.

As a QAM only describes how a specific group of users assigns quality to a certain product, several such models have to be used to describe quality assignment by different groups of users. As the product behaviour does not depend on the quality assignment, these different QAMs can be connected to one dynamic product model. In the case of tulip bulbs, two QAMs are defined, depending on the intended usage of the bulbs. One model is for bulbs destined for 'dry sales' directly to the consumer, whereas the other is for bulbs destined for the production of cut flowers ('forcing'). In both models quality assignment is described primarily as a linear function of bulb damage and bulb disease. The QAM for forcing gives more weight to disease, while the model for dry sales gives more weight to damage and other aspects of external appearance. Data about the way consumers evaluate bulb quality was obtained with a technique derived from conjoint measurement. This technique was also used to analyse the assignment of quality to ham [Steenkamp, 1987].

The development of a quality assignment model involves the *analysis of the quality notion* of a homogeneous group of users. Such a group of users of a product is assumed to have a common notion of quality for the product. The goal is to identify the quality attributes that are used in the quality notion, and to identify the relative weights assigned to each quality attributes and to combinations of quality attributes. All this is formulated in the quality assignment model (QAM). This step must be performed first, as through the identification of the relevant quality attributes a limitation is made on the product behaviour that has to be modelled.

3.3.2 The dynamic product model

The changes in the product properties represent the behaviour of the product in its environment. A *Dynamic Product Model* (DPM) describes how environmental conditions affect product behaviour. A DPM consists of several submodels, each describing a single process occurring in the product or an aspect of a process. The chilling injury model of [Tijskens *et al.*, 1994] is an example of how complex physiological behaviour can be decomposed into the constituting subprocesses, which are then described in separate submodels. The detrimental effect of radicals (process 1), generated both within and outside the product (process 2), is prevented by a radical scavenging system (process 3), which deteriorates at lower temperatures (process 4). The reaction rates of these four subprocesses all depend on temperature according to Arrhenius' law (process 5). Each subprocess describes a small but well-defined part of the behaviour of intrinsic product properties. The initial conditions and boundaries together with the generic model formulation explain and describe the various forms that chilling injury, as a process and as a property behaviour, can show.

The development of a dynamic product model consists of two steps. First, the quality attributes identified in the Quality Assignment Model are decomposed into intrinsic product properties. Each quality attribute is a combination of one or more intrinsic product properties. This step delivers a set of relevant intrinsic product properties that may influence quality of the product as perceived by the group of users. The result of this step is a mathematical formulation of these relations between quality attributes and intrinsic product properties. Second, the behaviour of the intrinsic product properties is modelled. In this step a set of processes and interactions between processes is constructed that together describe the behaviour of the relevant product properties. As the product behaviour is in fact the reaction of the product on its environment, by modelling the product behaviour also the external factors are identified that may affect the behaviour. Hence, the dynamic product model contains the mathematical formulations of the decompositions of the quality attributes into intrinsic product properties, and the mathematical formulations of the processes and their interactions that affect the intrinsic product properties.

3.3.3 The environment model

An *environment model* describes the changes in the environment of the product. In the case of non-packaged products, an environment model will be a series of environmental conditions at successive points in time, caused by the activities in the distribution chain under study.

If the product is packaged, the environment model describes the changes in the conditions inside the package (the micro-climate), as they are affected by physical processes such as diffusion of gases through the foil or the package, and by processes in the packaged product such as respiration, evaporation, and heat production.

The development of an environment model for packaged products involves the description of the physical processes that affect the conditions inside the package. These processes themselves are affected by material properties of the package, conditions outside the package, and the product inside the package.

3.4 Conclusions

The quality of a perishable product depends on the characteristics of the product itself, on criteria imposed by the user of the product on these characteristics, and on the availabilty and quality of alternative products. These three factors lead to the complex behaviour of quality observed during postharvest distribution of perishable products. The concepts of assigned quality and of acceptability of a product have been introduced, in which the effects of these three quality-determining factors are explicitly separated.

Assigned quality is an evaluation of the state of a product at a particular point in time. The product state is determined entirely by intrinsic product properties, which are in turn influenced by the environment. The product state is evaluated against quality criteria that reflect the needs and goals of the user of the product. The assigned quality therefore depends on three factors: the user of the product, the intrinsic properties of the product itself, and the interaction between the product and its environment. The assigned quality is important for product research, as it refers only to the changes in the product and to the criteria of a particular group of users. Product research is concerned with increasing the sales of a product by improving the product itself.

The acceptability of a product is an evaluation of the assigned quality in the context of extrinsic properties of the product such as its price, and in relation to other available products. Product acceptability includes a trade-off between price, availability and quality, whereas for the assigned quality itself, the price and the availability of other products are not relevant. Market research and consumer research focus on acceptability, and study how product sales can be increased using only economic instruments (e.g. price changes, advertising), hence without changing the product.

The changes in quality assigned to a product can be formalised in a quality change model. Such models consist of three submodels: a Quality Assignment Model describing the quality assignment by one user or one group of users that use the same quality criteria, a Dynamic Product Model describing the physiological behaviour of the product, and an Environment Model describing the changes in the environment of the product.

Using separate submodels has several advantages. First, the separation of product behaviour and quality assignment allows a description of the phenomena occurring in a product, independent of the user's attitude, and enables the same product model to be reused for different user groups. The separation also allows a clear description of the quality notions of users.

Separating the changes in the environment from the product behaviour has a similar advantage. However, in much of the literature on this topic, the processes occurring in the package and the behaviour of the packed products are combined into one model. Thus, such models directly link the product behaviour to the conditions outside the package. Separate modelling of the environment and of the product leads to a clearer conceptual description, and enables reuse of both the environment model and the dynamic product model.

A further advantage of using separate submodels is that different analysis and modelling techniques may be used for the three entities. The environment model describes physical processes, whereas the product model describes complex biochemical processes. Quality assignment has a psychological nature, for which empirical models may be more appropriate.

Chapter 4

Automated modelling of quality change

4.1 Introduction

The first part of this thesis focused on the concept of quality change of agricultural products. In Section 3.2 a definition of quality of perishable products was formulated. In Section 3.3 a quality change model (QCM) was defined. A quality change model consists of three submodels: a quality assignment model (QAM), a dynamic product model (DPM), and an environment model (EM). It was shown that this decomposition enables the modeller to describe the product behaviour independently of the usage of the product and of product environment.

The second part of the thesis focuses on the task of constructing such quality change models. In particular, we focus on the construction of the dynamic product model (DPM), that describes the part of the physiological behaviour that is relevant for the quality change of a perishable product. The QAM describes the way in which a consumer assigns quality to a product based on an inspection of product properties. Hence, the QAM specifies how the output of the DPM is transformed into assigned quality. The EM describes how the input of the DPM (i.e. the environment to which the product is subjected) changes. The construction of the QAM and the EM will not be discussed further in this thesis.

In the next chapters we will describe our approach to automate the construction of dynamic product models, and we will describe the implementation of this approach in the Desimal system. The Desimal system supports the modeller to decompose the observed complex behaviour of a product into a set of generic and primitive physiological processes. Subsequently a dynamic product model is constructed by selecting mathematical models for these processes. Below, it described what is meant by a generic process, when a process is called primitive, and why a quality change model should include all processes, also those processes that are not always observable.

A process that occurs in different products, or in different parts of one product can be modelled as a *generic* process. An example is evaporation, which occurs both in fruit and in cut flowers. In cut flowers, evaporation occurs both through the leaves and through the flower stem. A generic process may be a composition of several smaller processes corresponding with the underlying mechanisms of the generic process. If in all occurrences of the generic process the same underlying mechanism is used, then a generic composite model can be developed for the process. If a process has different underlying mechanisms in different products or in different parts of one product, then the process cannot be described by one generic composite model. In that case, one or more submodels in the generic model have to replaced by specialised models. As an example, colouring in tomatoes and colouring in apples involve different colour substances, so that at least different models are needed to describe the decomposition of the visible colour into the colour substances. The chemical reactions involved in these colouring processes may of course be described by generic models.

Processes can be decomposed into smaller processes. A process that cannot be further decomposed is called *primitive*. Decomposition of a process occurs at quantities involved in the process. A quantity that is influenced by exactly one process can be modelled as an internal quantity of the process, thereby hiding it from other processes and from the influences of other processes. A quantity that may be influenced by other processes has to be visible for the other processes. Hence, the complex process has to be decomposed in several smaller processes, in which the quantity is an influencing (input) or influenced (output) quantity. These processes are connected by interactions. An *interaction* relates an input quantity of one process to an output quantity of another process. In some cases, the modeller may decide not to decompose a complex process at a quantity that is influenced by other process. A reason might be that the other processes do not occur in the situations in which the phenomenon is studied. In that case, the quantity is modelled as an internal quantity in the complex process. If the model is later used to describe the phenomenon in a situation in which one of the other influencing processes does occur, then the model will produce an incorrect description of the phenomenon behaviour.

The resulting process decomposition often consists of processes that in the phenomenon under study occur with different intensities or process rates. Although the processes with a low intensity or process rate initially may not have an observable effect on the product behaviour, the effects of these processes may become observable after some time. Therefore, these processes are described in the dynamic product model, so that the moment at which the effects become observable can be predicted. For example, the water uptake through a flower stem is hampered if a certain part of the xylem vessels inside the flower stem are blocked by bacteria. Immediately after harvest, the xylem vessels are free of bacteria, but during the distribution period bacteria may enter the stem and grow, leading to the blocking of xylem vessels. By taking into account the bacteria growth and the vessel blocking processes from the beginning, the moment at which the water uptake becomes hampered can be predicted.

The proposed approach has several advantages. Firstly, the exhaustive decomposition of the product behaviour into generic and primitive processes results in a model that is a generic description of that part of the physiological behaviour. Processes in this behaviour that may be unobservable at the conditions at which the behaviour is analysed, will be described in the model, so that the model will not fail for situations where the effects of these processes may become observable. Secondly, by reusing generic processes and their corresponding mathematical models, the resulting DPM will consist of tested mathematical models, which improves the reliability of the DPM. Thirdly, by reusing generic processes and their corresponding mathematical models, the time and effort for developing a DPM is reduced.

4.2 Related research

The Desimal approach will use a combination of modelling support and automated model construction to assist in the construction of dynamic product models from a library of models for physiological processes. Modelling is usually supported by providing additional description levels to bridge the gap between the expert's knowledge about a certain class of problems and the available building blocks to solve these problems. Approaches to modelling support mainly focus on the contents of the additional description levels. Automated model construction aims at constructing the simplest simulation model that is sufficient for describing the behaviour of a system under study. This section reviews research on modelling support and on automated model construction.

4.2.1 Modelling support

Building a simulation model for a physiological phenomenon under study requires that the modeller translates his knowledge about the physiological behaviour of the product into a set of mathematical models of primitive processes. Such a large semantic gap between the expert's knowledge about a problem and the models in the mathematical model library can be bridged by introducing one or more additional description levels. Below, modelling support methods are reviewed for three different application domains, showing different forms of such intermediate description levels.

The Knowledge-Based Model Construction method of [Murray and Sheppard, 1988] supports a modeller in the construction of discrete event simulation models for queuing systems. First, a dialogue is held with the modeller to acquire a specification of the model to be constructed. In this dialogue domain knowledge and general simulation modelling knowledge is used. Second, this model specification is used together with general simulation modelling knowledge and knowledge of the target language to construct a discrete event simulation model in the target language.

In [Top, 1993] modelling is conceived as a 'process of making incremental and systematic assumptions'. In his *evolutionary modelling* method four description levels are identified in models for physical systems. First, the modeller assumes a decomposition into a set of *functional components*. Second, assumptions are made about the *physical processes* occurring in the functional components. Third, *mathematical equations* are assumed for each process leading to a mathematical model for the complete system. Fourth, values for the model parameters are assigned, which is the *model data* level. Each of these four description levels describes a different aspect of the system under study, that is not captured in the other levels. Hence, a complete model for that system contains descriptions for all four levels. Building a complete model is an iterative process for all four description levels. Each iteration consists of three modelling activities: specification of the requirements for the model, construction of the model, and assessment of the model. The assessment may result in a set of violated requirements. These violated requirements are input to a next iteration in which the set of requirements is modified, and a new model is constructed and assessed.

In [Mili, 1995] a framework for program library documentation is described that provides three so-called documentation templates: (1) for the problem to be solved, (2) for the programs in the library, and (3) for the applicability of a program to solve a particular problem. The third documentation template serves as an intermediate description level between the problem to be solved and the available programs in the library. It describes when a program is suitable for a problem, and how the program must be instantiated for that problem. It also describes when and why a program is not suitable to a problem and then points to other programs in the library that may be suitable. The framework has been applied to document financial and mass spectrometry libraries.

These methods illustrate different approaches to modelling support: Murray and Sheppard use a dialogue to construct a specification of the simulation model to be constructed. The information gathered in this dialogue is, however, not stored in a library. Top introduces two intermediate description levels between the expert's knowledge about a physical system and the mathematical model for that system: *viz.* functional components and physical processes. The library contains separate building blocks for each description level. Each building block is related to other building blocks in the adjacent description levels [Top *et al.*, 1995a]. Mili says the focus of her research to be 'mainly on helping users understand the usefulness and the limitations of the programs used, rather than on helping the users to select the correct program'. This is manifested by the application templates in the library that explicitly specify a mapping between the programs in the library and the problems to be solved.

The additional description levels introduced in these approaches serve to make explicit the assumptions and decisions that a modeller makes in developing a model. Explicit representation of modelling assumptions and reasoning about these assumptions is the key to automated model construction.

4.2.2 Automated model construction

In the field of qualitative reasoning [Forbus, 1990; Weld and de Kleer, 1990], several approaches to automated model construction have been presented. The automated modelling approaches construct models for answering queries about a given scenario. A *scenario* is a system or situation being studied, and is described by the physical structure of the system and (possibly) statements about the behaviour of the system (*e.g.* about the initial state, about steady-state assumptions, and about the operating range). The *query* poses a question about such a scenario, by specifying a set of quantities of interest that have to be explained by the scenario model, and a set of driving conditions. The model must explain the effects of these conditions on the quantities of interest. As an example, the query 'what happens to the flowering stage if the number of bacteria increases' has one driving condition (an increasing number of bacteria), and one quantity of interest (flowering stage).

To answer such a query a *scenario model* is constructed using a library of model fragments. The scenario model must be the simplest model that is sufficient to explain or predict the behaviour of the quantities of interest specified in the query. Several approaches have been described.

The *compositional modelling* method [Falkenhainer and Forbus, 1991; Falkenhainer and Forbus, 1992] provides basic concepts to automated construction of scenario models for the analysis of a system's short-term behaviour using a library of model fragments.

In [Iwasaki and Levy, 1994] an approach to automated construction of models for *simulation* is described. Given a query with the above structure, the constructed scenario model must explain how the quantities of interest change over time. As the simulation may go through any state satisfying the initial state specified in the query, the scenario model must contain all model fragments that can be active in any of these states, whether they are actually reached or not. A model fragment is active in a system state, if the input conditions and the operating conditions of the model fragment are satisfied in that state. The active model fragments form a simulation model that determines the next state of the system. This method differs from the method of Falkenhainer and Forbus, in that the scenario models contain all submodels that can be reached from the initial state. Part of the model selection task is performed during the simulation experiments to derive the applicable submodels at each state of the system.

In [Nayak, 1995] a method for automated model construction is described that aims at explaining a user-specified causal relation in a system under study. The causal relation specifies the behaviour that the user is interested in. For example, if the question refers to temperature changes around an electrical wire, also the thermal properties have to be described which are normally irrelevant for electrical wires.

The Tripel system [Rickel and Porter, 1994; Rickel, 1995] constructs qualitative models in the domain of plant physiology. This system uses a large botany knowledge base, containing causal relations between plant properties to specify physiological processes, and encapsulation relations on plant properties and on causal relations to express differences in levels of detail. Because many relationships in the botany knowledge base up to now cannot be modelled quantitatively, the level of detail reached in the qualitative models produced by Rickel's system cannot yet be reached in quantitative models.

More extensive reviews and discussions of approaches to automated model construction can be found in [Schut and Bredeweg, 1996; Xia and Smith, 1996]. The remainder of this section focuses on specification of the model fragments and on the requirements imposed on the constructed scenario models.

4.2.2.1 Specification and organisation of model fragments

Except for Rickel's approach, that uses a large knowledge base of causal relations between plant properties, the automated modelling approaches use a library of *model fragments*. Generally, a model fragment is a partial specification of the behaviour of an aspect of a system's behaviour, so that for a complete description of that aspect several model fragments are needed.

The automated model construction methods mentioned above use a similar specification of the model fragments, which was proposed in [Falkenhainer and Forbus, 1991]. The specification of a model fragment contains both the formulation of the model fragment and the assumptions underlying the formulation, and consists of four parts:

- *Participants* are the entities in the domain to which the model fragment applies. The participants are subject to conditions that define the *structural configuration* of the participants. The model fragment is only applicable to entities that satisfy these conditions.
- *Operating conditions* are restrictions on the values of the variables in the formulation of a model fragment. The operating conditions are used during the simulation to determine whether the model fragment can be activated, and to determine whether the values of the variables are valid.
- *Underlying assumptions* specify for which queries a model fragment may be relevant, and specify decisions about the formulation of the model

fragment, that were made during modelling.

- *Behaviour relations* are the qualitative or quantitative relations imposed by the model fragment between physical quantities of the participants, and can thus be seen as the formulation of the model fragment.

In [Gruber, 1993] a similar specification formalism for model fragments is presented, that consists of a specification of the *behavioural effects* described by the model fragment, *structural conditions* specifying which domain quantities are represented by the variables in the model fragment, *behavioural conditions* specifying inequalities between variables that must hold before the model fragment can be applied, and *model selection heuristics* specifying the components or processes for which a model fragment is relevant. The structural conditions and the behavioural conditions form the *applicability knowledge* of the model fragment, that specifies when during a simulation experiment the model fragments in the scenario model can be applied. The model selection heuristics correspond to the underlying assumptions in the model fragment specifications used by Falkenhainer and Forbus. This knowledge is used during the construction of a scenario model, if several model fragments are available for a component or process in the scenario, and one appropriate model fragment has to be selected.

To illustrate the kind of knowledge represented in the model fragments, Figure 4.1 displays a model fragment that defines some properties of a contained liquid. The participants of the model fragment are a container can and a liquid cl. The conditions for these participants specify the structural configuration for which the behavioural relations are valid: the can must be appropriate to contain a fluid, cl must be a liquid of a certain substance and must be in the can. The model fragment specification does not contain operating conditions, so that the model fragment is always applicable. The underlying assumption, expressed by the Consider condition, specifies that the model fragment is relevant for queries about cans containing a fluid. The behavioural relations determine the level and the pressure of the liquid in the can as a function of properties of the liquid and properties of the container.

Aspects of a system often can be modelled in different ways. Each possible way of modelling an aspect is described by a separate model fragment. For example, a model for a rechargeable battery must take into account the voltage of the battery and the charge-level of the battery[1]. The voltage produced by the battery can be constant or charge-sensitive. These alternatives are de-

[1] This is a simplified version of the example used in [Iwasaki and Levy, 1994].

```
ModelFragment ContainedLiquid( cl )
   Participants
      can Conditions Fluid-container( can )
      cl  Conditions Contained-liquid( cl ) And
                     Container-of( cl, can ) And
                     Substance-of( cl, sub )
   Operating-Conditions
   Assumptions
      Consider( Contained-Fluids( can ) )
   Behaviour
      level(cl) = (4 · mass(cl)) / (density(sub) · PI · square(diameter(can)))
      pressure(bottom(can)) = level(cl) · density(sub) · G
```

Figure 4.1: A model fragment defining the level and the pressure of a liquid in a container. Adapted from Fig. 2 in [Falkenhainer and Forbus, 1992].

scribed by separate model fragments. For the charge-level of the battery three alternative model fragments are available: one describing a constant charge-level, one describing a normal accumulation recharge, and one describing an accumulation recharge that also accounts for ageing of the battery.

Which model fragments are used for the voltage and for the charge-level depends on assumptions made about the voltage and the charge-level. For each aspect one assumption must be made, corresponding to one model fragment for that aspect. In the above example, the alternatives for the voltage and the alternatives for the charge-level are mutually exclusive, which means that only one model fragment can be applied in the model to describe the aspect. Such mutually exclusive model fragments (and the corresponding underlying assumptions) are grouped into *assumption classes*. In the construction of a scenario model, for each relevant aspect of the system one model fragment must be selected (in other words, a modelling decision must be made) from the assumption class of that aspect. Hence, a complete model for the battery needs one model fragment from the voltage assumption class and one model fragment from the charge-level assumption class.

The assumption classes contain model fragments with mutually exclusive underlying assumptions. Model fragments describing an aspect may also differ with respect to the operating region in which the behavioural relations are valid. In [Iwasaki and Levy, 1994] such model fragments describing one aspect under the same modelling assumptions, but in different operating regions, are grouped into *composite model fragments* (CMF). An assumption class then consists of CMFs with mutually exclusive modelling assumptions.

4.2.2.2 Requirements on the models

The automated modelling approaches impose requirements of sufficiency and simplicity on the constructed models.

The requirement of *sufficiency* or *adequacy* means that the scenario model only includes those aspects of the system under study that are needed to analyse the situation specified in the query with an appropriate level of detail. The consequence of the sufficiency requirement is that as soon as the query changes, a new scenario model must be constructed. The approaches differ on the implementation of the sufficiency requirement.

Falkenhainer and Forbus use a strict structural part-of hierarchy of all objects in the domain to determine the system boundary. Starting from the required scenario elements, the part-of hierarchy is traversed upwards, until the smallest system is found that comprises all required scenario elements. The objects in the domain outside this smallest system need not be considered.

Iwasaki and Levy use relevancy knowledge to determine the simplest adequate simulation model. For each quantity to be described by the simulation model, Iwasaki and Levy backward chain over the causal relations imposed by the model fragments to produce a list of model fragments that influence the quantity. From this list the simplest model fragment is selected that does not contradict the modelling assumptions for the partial model constructed so far. To determine the simplest model fragment, relevancy claims are used. Relevance claims are specified at the level of quantities, and are part of the model fragment specifications. A positive relevance claim specifies that a quantity is relevant for the model fragment. A negative relevance claim specifies that a quantity can be ignored. One model fragment is simpler than a second model fragment if a quantity that is relevant in the second model fragment, can be ignored in the first model fragment.

Nayak uses a function-based approach. The scenario model must explain the function of the device of interest. This function is specified in the query as an input/output relation, called an *expected behaviour*. Only the model fragments on this causal path are relevant. For example, depending on the expected behaviour a model for a wire may describe the electrical, the thermal or the mechanical properties of the material of the wire. The model is constructed as follows: first a comprehensive model is created that explains the specified expected behaviour. This model consists of the most complex model fragments for the relevant phenomena. This comprehensive model is simplified by replacing the complex model fragments with simpler model fragments, until selection of simpler model fragments results in an inadequate model.

The TRIPEL system [Rickel and Porter, 1994; Rickel, 1995] uses knowledge about the time scales at which changes in quantities occur. A differential influence has an explicit time scale. A functional influence represents an immediate change, and therefore does not have a time scale. Functional influences are always relevant. A differential influence is relevant only if the time scale of the influence is equal or smaller than the time scale of interest. Thus, if the quantities in the query change at a time scale of minutes, then the changes that take hours or days are considered to be negligible.

The requirement of *simplicity* ensures that the models describe the behaviour at the lowest level of detail that is needed to answer the query. The model fragments are ordered by a *simpler-than* relation based on the underlying assumptions of the model fragment. For example, a model for a chemical reaction assuming first-order kinetics is simpler than a model that assumes Michaelis-Menten kinetics. Usually, the simplest model fragment is applied, until it proves unsuitable to describe the behaviour. Nayak first selects the most complex model fragments, and then applies simplifications to find the simplest sufficient scenario model. Whereas Falkenhainer and Forbus arrange the model fragments in an assumption class in order of increasing complexity, and thus use an implicit measure of simplicity, Iwasaki and Levy base the measure of simplicity on the relevance of quantities, which is explicitly represented as *relevance claims*. A model fragment that assumes fewer quantities to be relevant is simpler. This *simpler-than* relation corresponds to the relations between models in the Graphs of Models approach [Addanki *et al.*, 1991].

4.3 The DESIMAL approach

In this section we introduce the DESIMAL method for automated construction of quantitative simulation models of postharvest processes occurring in agricultural products. We discuss where the methods described in Sections 4.2.1 and 4.2.2 have been used.

Figure 4.2 shows the three types of domain knowledge, and the two modelling tasks, that are used in the Desimal approach. The domain knowledge is divided into three levels: (1) qualitative models for physiological processes and for decompositions of quantities, (2) mathematical models, and (3) applicability knowledge specifying mappings between the mathematical models and the qualitative models.

The first modelling task is *Qualitative Process Analysis* (QPA), and uses the qualitative knowledge to construct a *Process Structure Graph*, which is a qualitative model for the phenomenon under study.

The second modelling task is *Simulation Model Construction* (SMC), and uses the applicability knowledge and the mathematical knowledge to construct a *Dynamic Product Model*, which is a quantitative simulation model for the processes and the interactions between processes described in the Process Structure Graph.

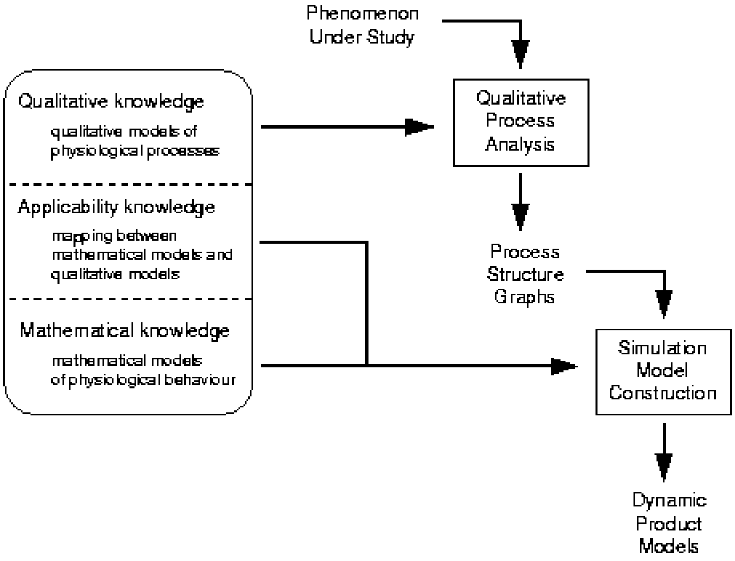

Phenomenon Under Study
Qualitative knowledge
qualitative models of physiological processes
Applicability knowledge
mapping between mathematical models and qualitative models
Mathematical knowledge
mathematical models of physiological behaviour
Qualitative Process Analysis
Process Structure Graphs
Simulation Model Construction
Dynamic Product Models

Figure 4.2: Overview of the Desimal approach showing the three levels of the Desimal library on the left, and the two modelling tasks on the right.

4.3.1 Representation of the domain knowledge

Automated model construction is only possible if apart from the object-level knowledge of the available models (the model formulation and the model variables) also meta-level knowledge about the applicability of the models for the problem under study is represented. In the approaches to automated model construction, reviewed in Section 4.2.2.1, both the object-level and the meta-level knowledge are represented in the model fragment specifications. The **participants** and the **behaviour relations** specify the object-level knowledge about the model formulation. The **assumptions** and the **conditions** of the participants specify meta-level knowledge about the relevancy of the model fragments to questions of interest.

Our approach is to separate these types of knowledge into three separate knowledge levels, that together form the library used in the modelling tasks of the Desimal approach (see Figure 4.2):

- A *qualitative knowledge* level, consisting of qualitative models for physiological processes and for decompositions of aggregate quantities into subquantities.
- A *mathematical knowledge* level, containing the mathematical models that are the building blocks for the quantitative simulation models.
- An *applicability knowledge* level between the qualitative and mathematical knowledge levels, that specifies mappings between the qualitative models and the mathematical models. This knowledge is used to select mathematical models for the processes and decompositions in a process structure graph.

4.3.2 The model construction task

The right side of Figure 4.2 shows the two modelling tasks of the Desimal approach. The model construction task in Desimal starts from a specification of the *phenomenon under study*, consisting of the quantities of interest, of which the behaviour over time must be explained, and a set of exogenous quantities, serving as the inputs of the simulation model.

The first modelling task is *Qualitative Process Analysis* (QPA). In this task one or more *Process Structure Graphs* are constructed for the phenomenon under study. Each process structure graph describes an adequate decomposition of the phenomenon under study into the generic processes represented by the qualitative models in the qualitative knowledge level of the Desimal library.

The second modelling task is *Simulation Model Construction* (SMC). In this task a *Dynamic Product Model* is constructed for each process structure graph. A dynamic product model consists of a set of mathematical models and an explicit simulation-control component that specifies when these models have to be used during a simulation experiment. This modelling task uses the applicability knowledge and the mathematical knowledge from the Desimal library. The mathematical knowledge level contains the mathematical models which are the building blocks of the dynamic product models. The applicability knowledge level provides the knowledge that is needed to select the appropriate mathematical models for the processes and decompositions in the process structure graph.

Qualitative Process Analysis and Simulation Model Construction both involve reasoning about the reusability of models in a model library. Component reuse in general involves three steps: retrieval, evaluation, and adaptation to fit new applications, see e.g. [Penix and Alexander, 1995]. Retrieval involves the selection of the components that are applicable for the problem under study. The goal of the evaluation step is to determine the suitability of the component for the problem under study. In the adaptation step, the necessary changes are made to the component so that the component can be reused for the problem under study. In the subtasks of the Desimal method, these steps have the following meanings.

Retrieval involves the selection from the library of the building blocks that are applicable for the phenomenon under study.

- In QPA, this subtask retrieves all qualitative models that describe an (indirect) effect of a process on a quantity of interest, and retrieves all qualitative models that represent how a quantity is decomposed into more detailed subquantities.
- In SMC, this subtask retrieves all mathematical models that are applicable to processes or quantity decompositions in the process structure graph developed for the phenomenon under study.

Evaluation involves the restriction of the comprehensive models derived in the retrieval subtask to models that are adequate for the phenomenon under study.

- In QPA, this subtask selects from the comprehensive process structure graphs those processes that represent an (indirect) effect of one of the exogenous quantities in the specification of the phenomenon under study on a quantity of interest. Only those quantity decompositions are selected that include detailed quantities that are needed to relate a quantity of interest to an exogenous quantity.
- In SMC, this subtask determines consistent model sets from the retrieved mathematical models. Each consistent model set is a simulation model that is adequate to describe the behaviour of the processes in a certain state of the simulation experiment.

Adaptation is not allowed in the DESIMAL approach. The qualitative models and the mathematical models in the DESIMAL library are indivisible building blocks.

The adequacy requirement applied in the DESIMAL method uses the causal dependencies between the product properties. Selecting the system boundaries by means of a part-of hierarchy of physiological processes, like the component hierarchy used by Falkenhainer and Forbus, is not possible. Such a hierarchy cannot be created as yet, because the interactions between the processes are to a large extent unknown. In fact, the aim of the DESIMAL approach is to assist the modeller in the investigation of interactions between physiological processes.

Selecting the relevant behaviour of the phenomenon under study by means of time-scale information, as is done in the TRIPEL system, is not possible. The simulation models constructed by DESIMAL are applied in the analysis of the behaviour of perishable products during long-term distribution chains, so that simulation period will be much longer than the time-scale at which the quantities change. For example, if the modelling question only contains quantities that change at the scale of minutes and the simulation model is used to analyse the behaviour during a period of five days, changes with a scale of hours and days will be important as well. These changes would be left out in the TRIPEL system.

4.4 Discussion

The approaches to automated model construction reviewed in Section 4.2.2 produce the simplest sufficient models for a given query. These models are constructed for testing hypotheses about the *behaviour* of the system under study. This justifies the automatic selection of the simplest possible model. However, the DESIMAL approach aims at supporting the modeller in the decomposition of the phenomenon under study into a set of generic or fundamental processes. The alternative process decompositions are presented to the modeller, who then selects the suitable decomposition. In this way, the modeller can test hypotheses about the appropriate *process decomposition* for the phenomenon under study.

The DESIMAL approach differs from the TRIPEL system [Rickel and Porter, 1994; Rickel, 1995]. Whereas TRIPEL system constructs a phenomenological model from a large knowledge base of causal relations between plant properties, the Qualitative Process Analysis task of the DESIMAL approach uses a library of generic or fundamental processes to decompose the phenomenon under study into a set of fundamental processes.

Compared to the method of [Iwasaki and Levy, 1994], the Simulation Model Construction task of the DESIMAL approach constructs compositional simulation models with an explicit control level.

The applicability knowledge used in the Simulation Model Construction task compares with the application templates in the approach of Mili. In her approach the application templates provide the intermediate description level between the problem under study and the available programs. However, from the viewpoint of the overall model construction task of the DESIMAL approach, the qualitative knowledge serves as the intermediate level between the expert's knowledge about the phenomenon under study and the collection of mathematical models.

Chapter 5

Qualitative Process Analysis

5.1 Introduction

As discussed in the previous chapter, the construction of a dynamic product model for a quality change phenomenon is composed of two separate automated modelling tasks. The construction starts from a modelling question, which is a specification of the phenomenon under study, consisting of quantities of which the behaviour over time must be explained (quantities of interest) and a set of exogenous quantities, serving as the inputs of the dynamic product model.

The goal of the first modelling task, *Qualitative Process Analysis* (QPA), is to find one or more qualitative models for the phenomenon under study, called process structure graphs. Each process structure graph represents an adequate decomposition of the phenomenon under study into a set of interacting generic processes. Qualitative Process Analysis uses the qualitative models of postharvest physiological processes, stored in the DESIMAL library. This is displayed in Figure 5.1, which is the upper part of Figure 4.2 discussed in Section 4.3.

Each process represented in the DESIMAL library affects one or more product properties and may itself be affected by other quantities, which can be product properties or external factors. Two processes interact if a quantity that affects one process, is itself affected by the other process. A qualitative model of a process represents which quantities are involved in the process, but hides the quantitative details of the process, which are not needed to find an adequate

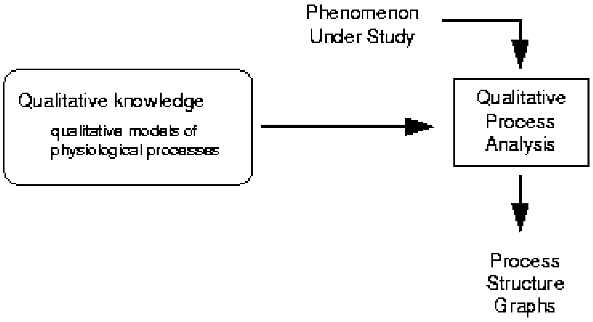

Phenomenon Under Study
Qualitative knowledge
qualitative models of physiological processes
Qualitative Process Analysis
Process Structure Graphs

Figure 5.1: Overview of the Qualitative Process Analysis task

decomposition of the phenomenon under study. The qualitative knowledge level allows the modeller to analyse what processes are involved in a quality change phenomenon under study, without bothering about the mathematical formulations for these processes.

Besides qualitative models of physiological processes, the qualitative knowledge in the Desimal library contains separate qualitative models describing how composite product properties are composed of more detailed product properties. These models are used when a phenomenon under study is modelled as a set of processes that involve product quantities at different levels of aggregation. For example, to model the changes in the firmness of a product, a model is needed that describes which chemical substances are responsible for the firmness, and that describes the relative importance of these substances for the overall firmness. With the knowledge in such a model, the processes affecting these substances can be selected, and the effects of these processes can be combined into the behaviour of the firmness of the product.

In Section 5.2 the formalism for the representation of this knowledge is presented. The construction of the process structure graphs is discussed in Section 5.3. In Section 5.4, Qualitative Process Analysis is compared to the techniques for the construction of qualitative models, that were reviewed in Chapter 4.

5.2 The domain knowledge

In this section, we propose a formalism, based on conceptual graphs [Sowa, 1984], to represent complex phenomena of quality change in perishable products as interactions between a number of predefined generic physiological processes.

The formalism is illustrated with the complex phenomenon of flower opening in cut rose flowers.

5.2.1 Example: opening of rose flowers

An important quality attribute of cut rose flowers is the flowering stage at the moment of purchase. The flowering stage is a measure for the appearance of the flowers that describes how far the flowers are opened. In this example, the flowering stage is described as a ranking on an integer scale between 0 (fully closed buds) and 6 (finished flowering). The changes in the flowering stage are modelled with an increasing discrete function. The flowering stage could also be described by the diameter of the flower bud. In that case, the changes in the flowering stage have to be modelled by a continuous function that increases to a maximum value corresponding to a fully opened flower, followed by a decrease representing the dropping of the petals. This will be not be further elaborated.

The opening of the rose buds is a complex process in which the water uptake through the xylem vessels in the flower stem plays an important role. The water uptake becomes hindered if a certain part of the xylem vessels inside the flower stem become blocked. Vessel blocking may be caused by growth of bacteria inside the stem, or by air bubbles that enter the vessels when the flower stems are kept dry. The growth of bacteria is an autocatalytic process with a rate that depends on temperature.

Previous exposure to bacteria or dry storage may cause the flower buds to open partially, even if the flowers are subsequently kept at optimal conditions. Therefore, the blocking of xylem vessels is assumed to have a negative effect on the maximum reachable flowering stage rather than on the rate of the flower opening process. This rate is only dependent on temperature, and can be modelled with the Arrhenius function for rates of chemical reactions [Chang, 1981]. The flower opening process itself is modelled as a limited exponential growth process.

5.2.2 Types of knowledge

The above description of the phenomenon of flower opening illustrates the reasoning involved when decomposing a quality change phenomenon into interacting generic processes. Domain knowledge is used about the behaviour of intrinsic product properties, about the decomposition of intrinsic product properties, about quality assignment relations between intrinsic product properties and quality attributes, about parameters of product properties, about modelling assumptions, and about applicable quantitative models. Below, these types of domain knowledge are elaborated, and for each type of domain knowledge examples are given from the modelling case of the opening of rose flowers.

- *quality assignment relations between product properties and quality attributes*
  This type of knowledge is specific to the task of modelling quality change. As described in Chapter 3, quality attributes are perceived combinations of one or more intrinsic product properties. The quality assignment relations specify which intrinsic product properties are perceived and evaluated to assess a quality attribute.

  > Example: the flowering stage is a quality attribute that is the perception of the flower bud diameter and the condition of the flower, such as damage, wilting, and necrosis, which are intrinsic product properties.

- *behavioural knowledge about product properties*
  This type of knowledge includes causal relations between intrinsic product properties and external factors, and knowledge about which processes are responsible for the behaviour of a product property. The behaviour of quality attributes is modelled through the behaviour of the intrinsic product properties that are perceived and evaluated for the quality attribute.

  > Example: the blocking of the xylem vessels is caused by bacteria in the flower stem, and by air bubbles in the flower stem.

- *parameters of product properties*
  The product properties may have parameters such as a maximum or minimum value, and a threshold value. These parameters are often

affected by other product properties or external factors. To capture this behaviour, the parameters are modelled as intrinsic product properties.

> Examples: the flowering stage has a maximum reachable value that is affected by the number of blocked vessels in the flower stem; the number of blocked vessels has a threshold that must be exceeded before the number of blocked vessels affects the water uptake through a flower stem.

- *decompositions of product properties*
  This type of knowledge is used to select the level of detail that is appropriate for describing the behaviour of the phenomenon under study. Differences in the level of detail are represented at the level of product properties. The DESIMAL library contains knowledge about the decomposition of a product property into more detailed product properties.

  > Example: the number of blocked vessels is a composition of the number of vessels blocked by bacteria and the number of vessels blocked by air bubbles in the flower stem.

  Both the decomposition of product properties and the quality assignment relations concern the representation of differences in the level of detail. Consequently, a quality assignment relation between a quality attribute and the perceived intrinsic product properties is modelled as a decomposition of the quality attribute. Hence, the colour of a tomato is a composition of the amounts of the colouring substances. By treating the quality assignment relations as differences in level of detail, a quality attribute can either be described as a black box quantity, or in terms of the perceived product properties.

- *modelling assumptions*
  An important type of knowledge are the modelling assumptions. A *modelling assumption* relates a specific process to a generic process, or states that a product property is assumed to be affected by a certain generic process.

  > Examples: the growth of bacteria is assumed to be an autocatalytic process; the flower opening is modelled as a limited exponential process governed by the amount of blocked vessels.

- *applicable quantitative models*
  The above description of the phenomenon of flower opening also contains

knowledge about the mathematical models that can be used to describe the behaviour of the quantities, *e.g.* the rate of flower opening can be modelled by an Arrhenius function. This type of knowledge is used only in the Simulation Model Construction task, rather than in the current Qualitative Process Analysis task, and will therefore not be discussed here.

5.2.3 Knowledge graphs

As a basis for the representation of the qualitative knowledge used in Qualitative Process Analysis, *knowledge graphs* [James, 1991; Willems, 1993] are used. One of the ideas underlying knowledge graphs is that one should limit the number of different relation types between domain concepts. This is in contrast to conceptual graphs or logical representations where the set of relation types is unlimited so that one can introduce new relation types whenever needed. Each knowledge graph represents the knowledge about a subject in the application domain. The nodes in a knowledge graph are concepts that are relevant for the subject. The arcs between these nodes represent cause-effect, part-of, and kind-of relationships between domain concepts. The arcs in a knowledge graph are labelled with abbreviations of the relation types: causal relations are labelled with CAU, part-of relations are labelled with PAR, and kind-of relations are labelled with AKO. A more complex conceptual structure can be formed by grouping concepts and relations into a *frame* with its own name. Frames are in turn concepts in the knowledge graph, and can be related to other concepts in the knowledge graph. To specify that a concept belongs to a frame, FPAR relations are used. In a graphical representation of a knowledge graph, the FPAR relations are kept implicit, and the frames are represented by boxes around the subgraphs.

In DESIMAL, knowledge graphs are used to represent the postharvest physiological behaviour that underlies the quality change of perishable products. Each knowledge graph in the DESIMAL library is a qualitative model for one quality change phenomenon of a perishable product. The concepts in these knowledge graphs are the quantities involved in the phenomenon, which are quality attributes, intrinsic product properties and external factors. A knowledge graph in the DESIMAL library may contain two types of frames: *process frames* to represent physiological processes and *decomposition frames* to specify decompositions of quantities. The frames are building blocks of the knowledge graphs in the DESIMAL library. Below the contents and the representation of the two types of frames are further elaborated.

5.2.3.1 Process frames represent physiological behaviour

A *process frame* represents the behaviour of a physiological process at one level of detail. The arcs between the quantities in a process frame are causal relations. We distinguish two types:

- dcau(source, target)
  The DCAU relation represents a differential causal relation. Hence, the value of the source quantity determines the rate of change (first derivative) of the target quantity. DCAU relations correspond with influences in Qualitative Process Theory (QPT) [Forbus, 1984], and with the differential influences in [Rickel, 1995]. The differential causal relations represent a differential equation for the target quantity on the mathematical level.

- fcau(source, target)
  The FCAU relation represents a functional causal relation. Hence, the value of the source quantity directly determines the value of the target quantity. FCAU relations correspond with qualitative proportionalities in QPT, and with functional influences in [Rickel, 1995]. The functional causal relations represent an algebraic equation for the target quantity on the mathematical level.

Figure 5.2 shows two examples of process frames that are used to model the opening of rose flowers. The process frame BacteriaBlocking represents the process of vessel blocking caused by bacteria. The DCAU relation between ConcBacteriaWater and ConcBacteriaStem represents the bacteria flow from the vase water into the flower stem. A DCAU relation is used, because the bacteria concentration in the water determines the bacteria flow rate into the stem, which in turn determines the increase of the bacteria concentration in the stem. The second relation in this process frame represents the vessel blocking. Here, an FCAU relation is used between ConcBacteriaStem and BacteriaBlockedVessels, because the number of vessels blocked by bacteria is proportional to the bacteria concentration in the stem.

The process frame FloweringObstruction represents the effect of vessel blocking on the maximum reachable flowering stage, which is represented by the

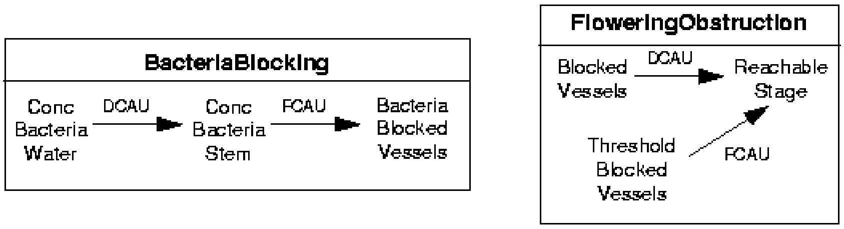


Figure 5.2: Two process frames describing processes involved in the opening of rose flowers. The left-side frame represents the blocking of vessels in the flower stem by bacteria. The right-side frame represents the effect of vessel blocking on the maximum reachable flowering stage.

quantity ReachableStage. This quantity depends on the amount of water taken up through the flower stem. The number of blocked vessels (represented by BlockedVessels) determines the maximal flow of water through the flower stem, and therefore, has a differential causal relation (DCAU) with ReachableStage. Vessel blocking has an effect on the maximal reachable flowering stage, only if the number of blocked vessels exceeds a certain threshold value. This is modelled with a functional causal relation from the property ThresholdBlockedVessels to ReachableStage.

The quantity ThresholdBlockedVessels is a parameter of the product property BlockedVessels. This fact is not represented in the knowledge graph, because the knowledge graphs in the Desimal library only represent the behavioural relations between the product properties.

5.2.3.2 Decomposition frames relate different levels of detail

The second type of frames are *decomposition frames*. Each decomposition frame represents the decomposition of a product property into product properties at a more detailed description level, and may only contain PAR relations:

- par(q_detailed, q_aggregate)
  The PAR relation represents that the quantity q_detailed is a part of the aggregate quantity q_aggregate. A PAR relation defines a boundary between the different levels of detail on which the behaviour of the aggregate quantity can be described.

The purpose of the decomposition frames is to provide the means to select the appropriate level of detail for describing the phenomenon under study. The decomposition frames represent structural relations between product properties, and are not intended to represent behaviour. As a consequence, causal relations are not allowed in decomposition frames.

Decomposition frames are also used to represent through which intrinsic product properties a quality attribute is evaluated. Hence, quality assignment, as described in Section 3.2.1, is modelled as a composition of intrinsic product properties into quality attributes. In this manner, a quality change model can describe the changes in a quality attribute either at the level of the quality attribute, disregarding the behaviour of the intrinsic product properties, or by modelling the behaviour of the intrinsic product properties.

As an example, colour is a quality attribute of tomatoes that is the perceived combination of the amounts of the colouring substances, chlorophyll and lycopene, which are intrinsic product properties. This quality assignment relation is represented by the decomposition frame Dec-TomatoColour in Figure 5.3.

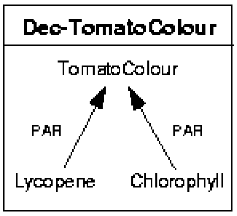


Figure 5.3: To assess the colour of tomatoes (quality attribute) the chlorophyll and lycopene amounts (intrinsic tomato properties) are perceived and evaluated. In the DESIMAL library, quality assignment is represented as a difference in the level of modelling detail.

In our approach we do not allow an aggregate quantity to have multiple decompositions into more detailed quantities. Hence, all constituent quantities of an aggregate quantity must be in one decomposition frame. A consequence of this is, that in many cases product-specific quantities have to be introduced, such as the quantity TomatoColour in the figure above, if a generic quantity has different decompositions into more detailed quantities in different products.

If a quantity depends on two or more other quantities, then it has to be decided whether the dependency has to be modelled by a set of PAR relations or by a set of causal relations. A set of PAR relations has to be used if the quantities are on different levels of detail, and the behaviour can either be described in terms of the quantity at the lower level of detail, or in terms of the more detailed quantities. The quantity at the lower level of detail is a combination of all quantities to which the quantity has PAR relations. It is not allowed to use only a subset of the PAR relations. Causal relations have to be used if the relationships between the quantities are time-dependent. A quantity may be the influenced quantity of a number of causal relations in different process frames. For each influencing process the modelling task will decide whether or not the process is relevant in the phenomenon under study. Hence, in a model a quantity may only depend on a subset of the causal relations that are defined in the DESIMAL library.

In the figure above, the quantity TomatoColour is modelled as a composition of the colour substances, because TomatoColour is on a lower level of detail than the quantities Lycopene and Chlorophyll, and because it is not possible to model the quantity TomatoColour as dependent on only one of the colouring substances Lycopene or Chlorophyll. This would be allowed if the dependency were modelled with causal relations.

5.2.3.3 Frames are connected by the quantities

A decomposition frame relates product properties at different levels of detail, but does not specify the behaviour of the product properties. Similarly, a process frame specifies a physiological process, but does not specify other processes in which the affected quantities may be involved. To connect the frames in a knowledge graph, two types of relations are used at the level of the quantities in the frames:

- equ(quantity1, quantity2)
  The EQU relation specifies that quantity1 in one frame and quantity2 in another frame represent the same product property. The EQU relations are used to connect separate frames that together describe one complex physiological process. These separate frames then contain quantities, possibly with differing names, that represent the same property. The EQU relation explicitly represents this relationship.
- ako(q_specific, q_generic)
  The AKO relation represents that the specific quantity q_specific is

> (modelled as) a kind of the generic quantity q_generic. An AKO relation can represent *generalisation* knowledge, *e.g.* in Figure 5.4 the quantity ConcBacteriaStem is a kind of ConcBacteria. The AKO relations can also represent *modelling assumptions*, *e.g.* the relation ako(ConcBacteria,AutoCatalyst) represents that the bacteria concentration is assumed be a kind of AutoCatalyst, thereby specifying that the growth of bacteria is assumed to be an autocatalytic process.

The EQU relations from the decomposition frame Dec-BlockVessels in Figure 5.4 specify that AirBlockedVessels and BacteriaBlockedVessels in this frame are the same as in the process frames AirBlocking and BacteriaBlocking, respectively.

The process frame BacteriaBlocking contains AKO relations from the quantities ConcBacteriaWater and ConcBacteriaStem to the generic quantity ConcBacteria to specify that both the bacteria concentration in the stem and the bacteria concentration in the vase water are influenced by a separate bacteria growth process. In this case AKO relations are used, because ConcBacteriaStem and ConcBacteriaWater are more specific quantities than the quantity ConcBacteria, and separate instances of the process frame BacteriaGrowth have to be created for the two bacteria concentrations.

The EQU and AKO relations are the only relations that are allowed between frames in the DESIMAL library. In the knowledge graph formalism as presented in [James, 1991], frames are concepts in the graph and thus can be related to other concepts. In the DESIMAL library, however, relationships between frames, such as decompositions of processes and generalisations of processes, are represented at the level of quantities.

Decomposition of a complex process into subprocesses is modelled by introducing intermediate quantities that are affected by the subprocesses. The quantity that is affected by the complex process is modelled as a composition of the intermediate quantities. For example, vessel blocking in flower stems is a complex process that is the composite of vessel blocking caused by bacteria and vessel blocking caused by air bubbles. The decomposition frame Dec-BlockedVessels in Figure 5.4 shows that the decomposition of the vessel blocking process is represented at the level of quantities: the quantity BlockedVessels is decomposed into the quantities BacteriaBlockedVessels and AirBlockedVessels, which are affected by the processes BacteriaBlocking and AirBlocking, respectively.

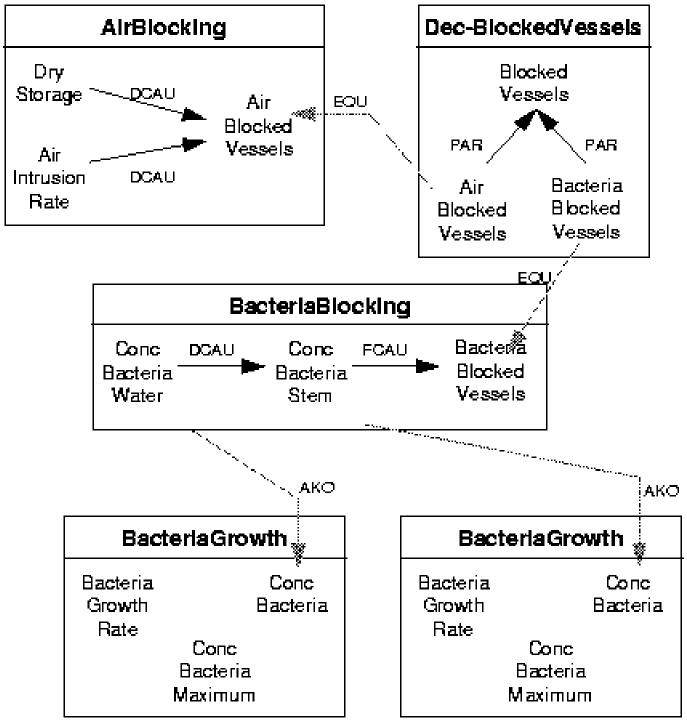


Figure 5.4: Knowledge graph representing that vessel blocking in a flower stem is caused by growth of bacteria in the stem and by air bubbles in the vessels. The bacteria concentration in the stem is also affected by bacteria flowing from the vase water into the stem. The bacteria growth in the stem and in the vase water are separate processes.

Just as for quantities, one process can be (modelled as) a kind of another process. This relationship can be a generalisation relationship (*e.g.* an enzymatic reaction is a chemical reaction), or a modelling assumption for a specific process (*e.g.* FlowerOpening is modelled as a kind of LimitedExponential process). These relationships between two processes are in our approach modelled by AKO relations between the quantities in the two process frames. These AKO relations specify the mapping of the quantities in the generic frame on the quantities in the specific frame. In this way, the role of the quantities in a process is explicitly represented. For example, the AKO relations in Figure 5.5 between the quantities in the BacteriaGrowth and AutoCatalysis frames represent that the growth of bacteria can be modelled as an autocatalytic process, in which ConcBacteria is the AutoCatalyst. When using AKO relations at the level of frames this mapping of the quantities could no longer be represented explicitly.

5.2.3.4 Properties of the quantities

The quantities in a knowledge graph represent the product properties and the external factors involved in the modelled quality change phenomenon.

Quantities are identified by their name. Two quantities with the same name represent the same product property or external factor. Quantities do not have additional attributes, such as location or unit of measure. The unit of measure is quantitative information, and is therefore not represented in the knowledge graphs. If information about the location of a product property is needed, then this has to be included in the name of the quantity. For example, vessel blocking caused by bacteria involved the bacteria concentrations in the vase water and in the flower stem. These properties are represented by separate quantities ConcBacteriaWater and ConcBacteriaStem.

Two quantities with different names may represent the same product property. This is represented by an EQU relation between the quantities. In the knowledge graphs shown in this thesis, quantities in different frames that have the same name, are connected by EQU relations. However, these relations are not used in the model construction task.

If a knowledge graph contains multiple instances of one process frame or decomposition frame, then the quantities in those frames that represent different product properties have to be given a unique name. This is done automatically by the implementation of the model construction task. When a new process frame is inserted into the knowledge graph, each quantity in the process frame that does not have an EQU relation with a quantity already in the knowledge graph, is duplicated and given a unique name.

The quality change models constructed by the DESIMAL method are suitable to calculate the effect of varying external conditions on the quality change of the modelled product or products. It is, however, assumed that the environment around the modelled product or products is homogeneous at any time point. As a consequence, the quantities that represent external factors cannot be duplicated.

5.2.4 Knowledge graphs for generic processes

Qualitative Process Analysis uses a library of knowledge graphs for generic processes. Figure 5.5 displays the frames in the library that are used to construct a knowledge graph for the phenomenon of flower opening. These frames are described below.

ReactionKinetics and ReactionSetKinetics are process frames for generic processes that relate the rate of a chemical process to temperature. The frame ReactionKinetics represents the temperature dependency of the rate of one chemical reaction, represented by the quantity SimpleRate. The frame ReactionSetKinetics represents the temperature dependency of the ApparentRate of a chain of chemical reactions. In principle, the ApparentRate is a composition of several SimpleRate quantities. As this decomposition depends on the actual rate being modelled, the DESIMAL library does not contain a generic decomposition frame for ApparentRate.

The process frame LimitedExponential represents the generic process of limited exponential growth. This process is a qualitative representation of a limited exponential function. The process affects the generic quantity LE_Quantity, which ranges between the values of the quantities LE_Minimum and LE_Maximum, which are parameters of LE_Quantity.

The process frame AutoCatalysis represents the generic process of autocatalysis. The process affects the generic quantity AutoCatalyst, which has two parameters AC_Minimum and AC_Maximum, representing the minimum and maximum value of the AutoCatalyst, respectively.

The DESIMAL library also contains more complex processes that consist of several generic frames, and that can be reused as one complex process. An example is the process frame BacteriaGrowth. Growth of bacteria in flower stems is modelled as an autocatalytic process, as is represented by the AKO relations between the quantities of the process frame BacteriaGrowth and AutoCatalysis. The BacteriaGrowthRate is a specialisation of the quantity ApparentRate in the generic process frame ReactionSetKinetics.

A chemical reaction is also modelled as a complex process, that consists of three process frames. The frame ChemicalReaction specifies the quantities involved in a chemical reaction. The quantity SimpleRate depends on temperature as represented in the process frame ReactionKinetics. The decrease of the quantity ConsumedReactant is represented by the process frame DecayReaction. The increase of the quantity ProducedReactant is described by

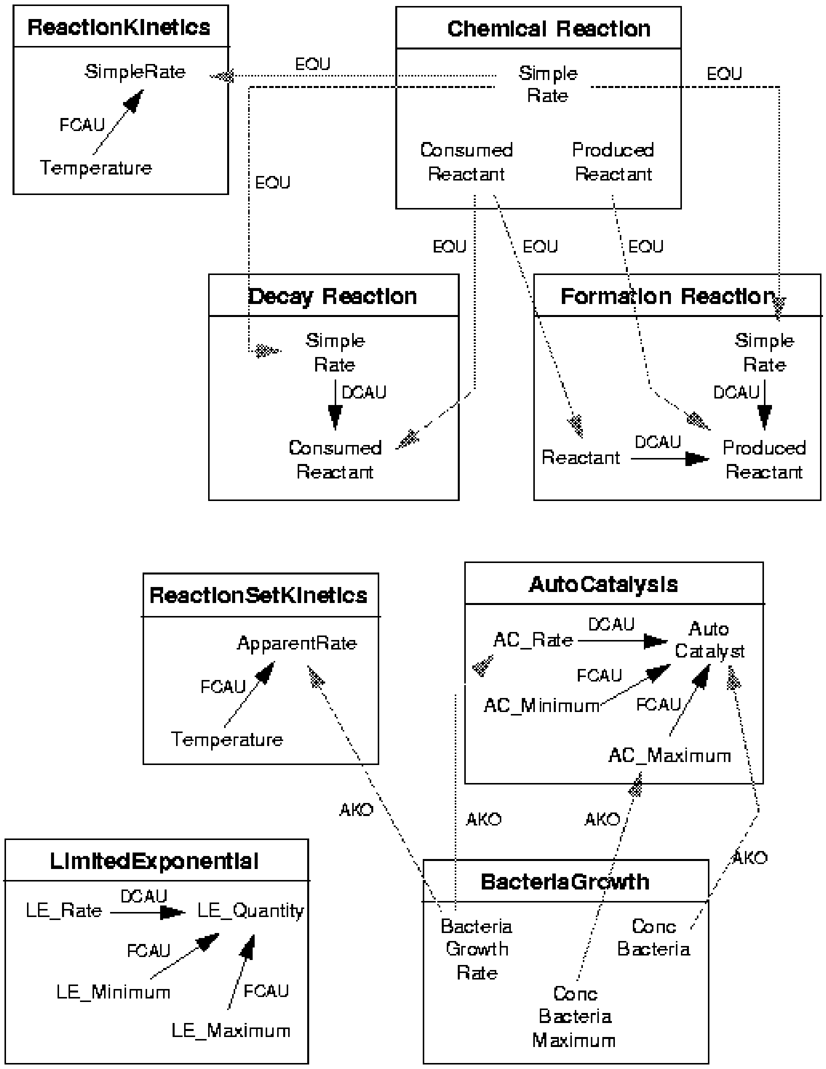


Figure 5.5: The qualitative knowledge in the DESIMAL library contains generic process frames that are reused to model postharvest physiological processes. Process frames in the library can be related to represent more complex generic processes.

the process frame FormationReaction. The quantities in the three frames are the same, as is represented by the EQU relations. Separating the degradation and formation processes enables the modeller to include only the effect on one of these quantities. Complex chemical reactions can be modelled by using decomposition frames for specialisations of the quantities ConsumedReactant and ProducedReactant.

5.2.5 Knowledge graph for the opening of rose flowers

The frames for the generic processes described above are used to develop a knowledge graph for the phenomenon of flower opening described in Section 5.2.1. The product properties involved in this phenomenon are grouped in the process frame FlowerOpening in the knowledge graph displayed in Figure 5.6.

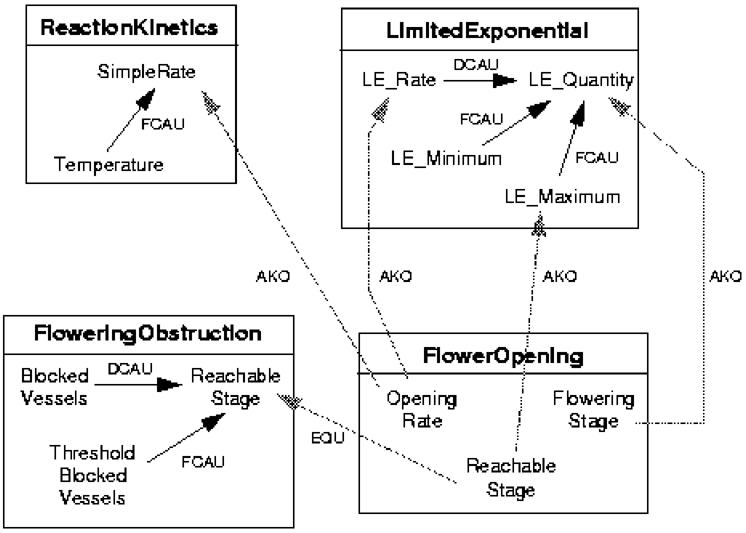


Figure 5.6: Knowledge graph for the opening of rose flowers, representing that the flowering stage can be modelled with a limited exponential function.

The behaviour of the quantities in this frame is specified by AKO relations to generic process frames from the library. The AKO relation from the quantity OpeningRate to the quantity SimpleRate in the generic frame ReactionKinetics specifies that the rate of flower opening is assumed to be monotonically changing with temperature. The other AKO relations specify that the behaviour of the flowering stage of cut roses is modelled with a limited exponential function.

The process frame FloweringObstruction is specific for the phenomenon of flower opening, and represents that the maximum reachable flowering stage (ReachableStage) depends on the number of blocked vessels. The product property ThresholdBlockedVessels is a threshold value for the visible negative effect of vessel blocking on the ReachableStage. The FCAU relation between these quantities represents that the maximum reachable stage changes if the threshold value changes.

The processes that are involved in the blocking of flower stem vessels are shown in Figure 5.4 on page 64. The decomposition frame Dec-BlockedVessels demonstrates that the decomposition of the complex process of vessel blocking is modelled at the level of the quantities rather than at the level of processes: the total amount of blocked vessels (BlockedVessels) is composed of the number of vessels blocked by bacteria (BacteriaBlockedVessels) and the number of vessels blocked by air bubbles (AirBlockedVessels). The process frame AirBlocking is an empirical model of the part of the vessel blocking that is caused by air entering the flower stem during periods of dry storage. The process frame BacteriaBlocking is a model for the part of the vessel blocking that is caused by the bacteria concentration in the stem (ConcBacteriaStem). The latter quantity is influenced by three processes: (1) growth of bacteria in the stem, (2) transport of bacteria out of the vase water into the stem, and (3) growth of bacteria in the vase water (ConcBacteriaWater). The transport process is represented as a DCAU relation between the quantities ConcBacteriaWater and ConcBacteriaStem. These two quantities are specialisations of the generic quantity ConcBacteria, that is influenced by the generic process BacteriaGrowth, stored in the process library (Figure 5.5).

5.3 The Qualitative Process Analysis task

Qualitative Process Analysis constructs one or more *process structure graphs* for a phenomenon under study. A process structure graph is a graph of interacting processes, in which each process is an instance of a process frame retrieved from the qualitative knowledge level in the DESIMAL library. Two processes interact if a property that affects the one process, is itself affected by the other process. A process structure graph may contain one or more product properties that are decomposed into more detailed product properties. The process structure graph then contains a decomposition frame that represents how the behaviour of the aggregate product property has to be described in terms of the detailed product properties. Note that a process structure graph will not contain decomposition frames, unless one or more product properties in the process structure graph are modelled as compositions of more detailed product properties.

A process structure graph only contains DCAU and FCAU relations, representing a piece of behaviour, and PAR relations, representing decompositions of aggregated properties into detailed product properties. The AKO and EQU relations represent structural knowledge, such as modelling assumptions and specialisations. These relations are used in the DESIMAL library to represent relationships between frames. The process structure graphs do not contain these relations. If a quantity has an AKO relation or an EQU relation to a second quantity, then the relations that hold for the second quantity also apply to the first quantity. This not only holds for the causal and PAR relations, but also for the AKO and EQU relations, so that chains of modelling assumptions and specialisations can be specified.

Qualitative Process Analysis starts from a *modelling question* about the phenomenon under study. The modelling question consists of a set of quantities of interest and a set of exogenous quantities. The process structure graphs constructed by the Qualitative Process Analysis task describe the effects over time of the exogenous quantities on the quantities of interest. Section 5.3.1 describes how the modelling question is specified.

As described in Section 4.3, Qualitative Process Analysis involves reasoning about the reusability of the knowledge graphs in the DESIMAL library, and consists of a retrieval and an evaluation subtask. The retrieval subtask selects from the DESIMAL library the process frames that describe an (indirect) effect of a process on a quantity of interest. This subtask also retrieves the decomposition frames that represent how a quantity in one of the process frames is decomposed into more detailed subquantities. The result are one or more comprehensive process structure graphs that include all processes that (indirectly) affect the quantities of interest. The retrieval subtask is further described in Section 5.3.2.

The evaluation subtask simplifies the comprehensive process structure graphs derived in the retrieval subtask to process structure graphs that are adequate for the phenomenon under study. The resulting adequate process structure graphs only contain the process frames that represent an (indirect) effect of one of the exogenous quantities on a quantity of interest. Furthermore, the adequate process structure graphs only contain the decomposition frames that introduce the detailed quantities that are needed to relate a quantity of interest to an exogenous quantity. The evaluation subtask is further described in Section 5.3.3.

5.3.1 Specify phenomenon

The modelling question for Qualitative Process Analysis consists of a set of *quantities of interest*, a set of *exogenous quantities*, and a set of *irrelevant quantities*:

- *Quantities of interest.* Quantities that are expected by the modeller to be involved in the behaviour of the phenomenon under study. Quantities of interest determine which aspects of the phenomenon under study have to be described at what level of detail.
- *Exogenous quantities.* The input quantities of the process structure graphs to be constructed. The graphs must describe the effects of these quantities on the quantities of interest.
- *Irrelevant quantities.* Quantities that have to be excluded from the process structure graphs. The modeller can specify a quantity to be irrelevant, if the effects of this quantity can be neglected in the phenomenon under study, or to force an alternative description of the behaviour of the quantities of interest. This may be important when a quantitative simulation model that was constructed on the basis of an earlier process structure graph, proves to be inappropriate. To construct a different process structure graph for the phenomenon, the inappropriate quantities have to be excluded explicitly.

Usually, one or more external factors are used as exogenous quantities. The external factors are affected both by the environment (*e.g.* a package) and by processes occurring in the product (*e.g.* respiration and heat production). As was discussed in Section 3.3, the effects of the environment on the external factors have to be described in a separate Environment Model. Effects on the external factors of the processes occurring in the product have to be included in the process structure graphs, only if the product is in a confined space, such as a package, or if interactions between products have to be considered, such as in mixed loads. For phenomena studied in products in an unconfined space, the effects of processes occurring in the product on the external factors can be neglected.

The modelling question is in fact a representation of the experimental setup: the exogenous quantities are the quantities that are varied in the experiment; the quantities of interest are the quantities of which the behaviour is studied in the experiment. Therewith the modelling question also determines the system boundary of the process structure graphs and of the dynamic product models developed from these graphs: the quantities of interest determine what quantities must be described, and the exogenous quantities determine what effects on the quantities of interest are of importance to the modeller.

5.3.2 Retrieval of relevant frames

The goal of the retrieval subtask is to construct one or more comprehensive process structure graphs for the phenomenon under study. Each comprehensive process structure graph contains all chains of process frames that connect the quantities of interest to the exogenous quantities via a specific set of intermediate quantities.

All process frames that have a direct effect on a quantity have to be included in the process structure graph, unless the process frame is itself influenced by a quantity that is irrelevant according to the modelling question. Multiple process structure graphs with different sets of intermediate quantities are generated only, if any of the quantities of interest depends on a quantity that is a composition of more detailed quantities. The behaviour of such an aggregate quantity can be modelled in two ways: at the aggregate level, or as the composition of its constituent detailed quantities. These models are represented in separate process structure graphs. One process structure graph will contain the decomposition frame for the aggregate quantity and does not contain the process frames that directly influence the aggregate quantity. The other process structure graph will contain the process frames that directly influence the aggregate quantity, and does not contain the decomposition frame and the detailed quantities.

The retrieval subtask uses the algorithm displayed in Figure 5.7 to generate these comprehensive process structure graphs. The algorithm consists of the following steps:

- Iteratively, a quantity q is selected from the process structure graph

```
initialise the set of candidate process structure graphs
    with a graph consisting of the quantities of interest and the exogenous
          quantities specified in the modelling question.
foreach psg in the set of candidate process structure graphs:
    foreach  quantity q in psg that is not an exogenous quantity, and
                that is not dependent on other quantities in the psg:
        foreach decomposition frame df that decomposes q into subquantities
            a) create a copy psg2 of the current process structure graph.
            b) include the decomposition frame df in psg2.
            c) add psg2 to the set of candidate process structure graphs.
        end foreach
        if no decomposition frames found
        then
            foreach process frame pf that affects q
                include the process frame in the process structure graph psg.
            end foreach
        foreach frame f included in psg:
                foreach group of AKO relations between quantities in f and
                           quantities in a more generic frame f2:
                    a) create an instance of the frame f2;
                    b) add this instance to the process structure graph.
                    c) add this instance to the set of selected frames
                       to consider the AKO and EQU relations in this frame.
                end foreach
                foreach group of EQU relations between quantities in f and
                           quantities in another frame f2:
                    if the frame f2 is not included in the process structure graph
                    then
                       a)  add the frame f2 to the process structure graph.
                       b)  add this instance to the set of selected frames to
                           consider the AKO and EQU relations in this frame.
                    else
                       consider q1 in frame f and q2 in frame f2 to represent
                       the same product property.
                end foreach
        end foreach
    end foreach
end foreach
```

Figure 5.7: Algorithm to generate comprehensive process structure graphs.

under construction (psg), and the process structure graph is extended with frames that represent an effect on this quantity. The effects are found by backward chaining over the causal relations. To prevent the algorithm from entering an infinite loop when a quantity is part of a feedback loop, a quantity q can only be selected if the process structure graph does not have a relation from another quantity p to q. Exogenous quantities in the process structure graph will never be selected, as the effects on these quantities have to be ignored. The algorithm stops when all other quantities in the process structure graph are dependent on exogenous quantities.

- For each selected quantity, the DESIMAL library is searched for frames that represent an effect on the selected quantity. Both process frames and decomposition frames may be found.

  Each process frame is included in the current process structure graph, as a separate effect on the quantity that may be relevant for the phenomenon under study.

  Each decomposition frame is inserted in a separate candidate process structure graph, which is a copy of the process structure graph under construction. This candidate process structure graph represents one possible decomposition of the selected quantity into subquantities, and is further developed in parallel with the current process structure graph.

  Hence, if the DESIMAL library contains one process frame and two decomposition frames for an aggregate quantity, then in second step of the algorithm three candidate process structure graphs will be constructed, one containing only the process frame, and the other two containing only one of the decomposition frames. If the DESIMAL library contains only a decomposition frame for the aggregate quantity, then the algorithm will still construct two candidate process structure graphs. One graph will contain the decomposition frame, the other graph will describe the quantity as an exogenous quantity.

- Each of the frames may contain AKO and EQU relations that connect quantities in the frame with quantities in other frames in the process structure graph or in the DESIMAL library. All AKO relations between two frames are treated as a group. The same holds for the EQU relations.

  For each group of AKO relations and for each group of EQU relations from one frame to a second frame, an instance of the second frame is created and added to the process structure graph. Hence, the EQU relations are treated in the same way as the AKO relations, only the semantics of these relation types for representing knowledge about physiological processes differ.

The algorithm correctly includes feedback loops into the process structure graph. Suppose the quantity of interest y depends on another quantity x which in turn depends on y. First quantity y will be selected, because it is a quantity of interest. Searching the influences on y will introduce the quantity x and the causal relation from x to y. Next, the quantity x is selected, and influences on x are searched. This will introduce the causal relation from y to x. As the process structure graph already contains a causal relation influencing y this quantity will not be selected again to search influences for.

The number of candidate process structure graphs depends on the number of aggregate quantities and the number of available decomposition frames for these aggregate quantities. The candidate process structure graphs are generated in the second step of the algorithm. In the way this step is described in the algorithm, this step may lead to a combinatorial explosion of the number of candidate process structure graphs. For example, if the phenomenon under study has to be modelled with two aggregate quantities, then four candidate process structure graphs are generated. In general, if n aggregate quantities are involved, then 2^n candidate process structure graphs are generated.

This combinatorial explosion can be avoided in two ways. Firstly, a copy of the process structure graph can be created only if the DESIMAL library contains a process frame that influences the aggregate quantity at the aggregate level. If the DESIMAL library contains only decomposition frames for an aggregate quantity, then the aggregate quantity cannot be modelled at the aggregate level. In that case, the process structure graph in which the aggregate quantity is not decomposed becomes superfluous. For example, the quantity BlockedVessels is a composition of the quantities BacteriaBlockedVessels and AirBlockedVessels (see the frame Dec-BlockedVessels in Figure 5.4). As there are no process frames that affect this quantity at the aggregate level, the quantity BlockedVessels must be described in terms of the detailed quantities. The decomposition frame can be inserted into the process structure graph under construction, instead of in a copy of the process structure graph.

Secondly, if an aggregate quantity is encountered, then at first only the aggregate level description can be generated. The modeller has to decide whether to generate the process structure graph in which the aggregate quantity is decomposed into its subquantities.

5.3.3 Selection of adequate process structure graphs

The process structure graphs constructed in the retrieval subtask contain all process frames that represent effects on the quantities of interest, and all decomposition frames that decompose a quantity in the process structure graph into more detailed quantities. For phenomena studied in a product in a confined space also the process frames are included that represent effects of the product on the external factors.

As the goal of Qualitative Process Analysis is to devise qualitative models describing the effects of given exogenous quantities on one or more quantities of interest, the constructed process structure graphs may contain frames that are irrelevant for the phenomenon under study. In the evaluation subtask, each comprehensive process structure graph is simplified to a process structure graph that is adequate for the phenomenon under study.

A process structure graph is adequate for the phenomenon under study if the following conditions are satisfied:

- The process structure graph must contain all quantities of interest, but may leave out exogenous quantities that do not affect any of the quantities of interest. However, if the product in which the phenomenon is studied is in a confined space, the process structure graph has to contain the external factors that are affected by the product.

- The behaviour of each aggregate quantity in the process structure graph must either be described at the aggregate level, or in terms of the detailed quantities of which the aggregate quantity is composed. In the latter case, the process structure graph must contain a decomposition frame that represents how the aggregate quantity is decomposed into the detailed quantities.

- The process structure graph must only contain process frames that represent an (indirect) effect on a quantity of interest and that themselves are (indirectly) affected by an exogenous quantity. If the product in which the phenomenon is studied is in a confined space, the process structure graph has to contain the process frames that represent an effect on the external factors affected by the product.

- The process structure graph must be connected, to prevent the description of unrelated pieces of behaviour.

The second condition specifies that an aggregate quantity cannot be both affected by a process, and decomposed into its subquantities in the same process structure graph. These are two alternative ways of modelling of an aggregate quantity, that have to be represented in two separate process structure graphs.

Both the aggregate level description and the detailed level description are considered to be adequate models for an aggregate quantity. This is in contrast with the approaches to automated modelling reviewed in Chapter 4. In these approaches each element in the studied system is described at an aggregate level, until this description proves inadequate to account for the observed behaviour of the element. In that case, another assumption is selected from the assumption class for the aggregate quantity. The new assumption implies a change of the model, so that the behaviour of the element is described at a more detailed level. Besides the assumption classes, [Falkenhainer and Forbus, 1991] use granularity assumptions to specify the level of detail that is required in the constructed model. Each granularity assumption specifies an element that has to be incorporated in the model, and furthermore specifies that that element cannot be decomposed into its sub-elements.

The third condition states that processes that (indirectly) cause changes in a quantity of interest, but that are themselves not affected by exogenous quantities in the phenomenon under study, are considered to be irrelevant. Hence, it is this condition that leads to a simplification of the comprehensive process structure graphs. The processes that are not affected by exogenous quantities can be ignored, because the process structure graphs are constructed for a specific modelling question. The quantities in the modelling question determine which influence paths are relevant. The other influences on the quantities of interest are irrelevant for the modelling question. If, according to the modeller, these influences are relevant, then the modelling question has to be changed.

The comprehensive process structure graphs are simplified by determining the influence paths from the exogenous quantities to the quantities of interest. This is done by a traversal over the DCAU, FCAU and PAR relations in the comprehensive process structure graph, starting from the exogenous quantities and ending with the quantities of interest. The frames that are not on these paths are considered to be irrelevant. In this manner, all effects of the exogenous quantities on the quantities of interest are kept in the process structure graph. Also all effects of processes on the external factors are kept. Although this satisfies the third condition given above, a modeller may want to leave out some of the effects represented in the adequate process structure graph.

For example, the effects on the external factors can be neglected, if the product in which the phenomenon is studied is in an unconfined space. As a second example, if a phenomenon under study includes an enzymatic reaction, then the comprehensive process structure graph constructed for that phenomenon will always contain a denaturation process for that enzyme. If the temperature is specified as an exogenous quantity, the evaluation task will consider the denaturation process to be relevant. The evaluation task cannot decide whether or not the denaturation process is indeed relevant for the phenomenon under study, because this depends on the situations in which the phenomenon is studied.

The above examples show that after the evaluation task, a modeller may further simplify the adequate process structure graphs by leaving out certain processes that in the modelling task are considered to be relevant.

5.3.4 Process structure graph for bacteria growth

The subtasks of the Qualitative Process Analysis task are illustrated with the construction of a process structure graph for the generic bacteria growth process. The knowledge graph of this generic process is displayed in Figure 5.5. For this example, suppose that the modelling question specifies ConcBacteria as the quantity of interest, and Temperature as the exogenous quantity.

The steps in the construction of the process structure graph are displayed in Figure 5.8. To find the process structure graph for this modelling question, Qualitative Process Analysis starts with a process structure graph that contains only the quantities ConcBacteria and Temperature. The initial process structure graph is displayed in Figure 5.8 (A).

- First, the process frame BacteriaGrowth is included, as this is the only process frame in the DESIMAL library that influences the quantity of interest ConcBacteria (Figure 5.8 (B)).
- Next, the AKO relations from this process frame to the process frame AutoCatalysis are followed. Instantiation of this process frame introduces an additional quantity AC_Minimum_1, and adds the relations between the quantities already included in the process structure graph (Figure 5.8 (C)).
- The quantity BacteriaGrowthRate also has an AKO relation to the process frame ReactionSetKinetics. Instantiation of this process frame introduces

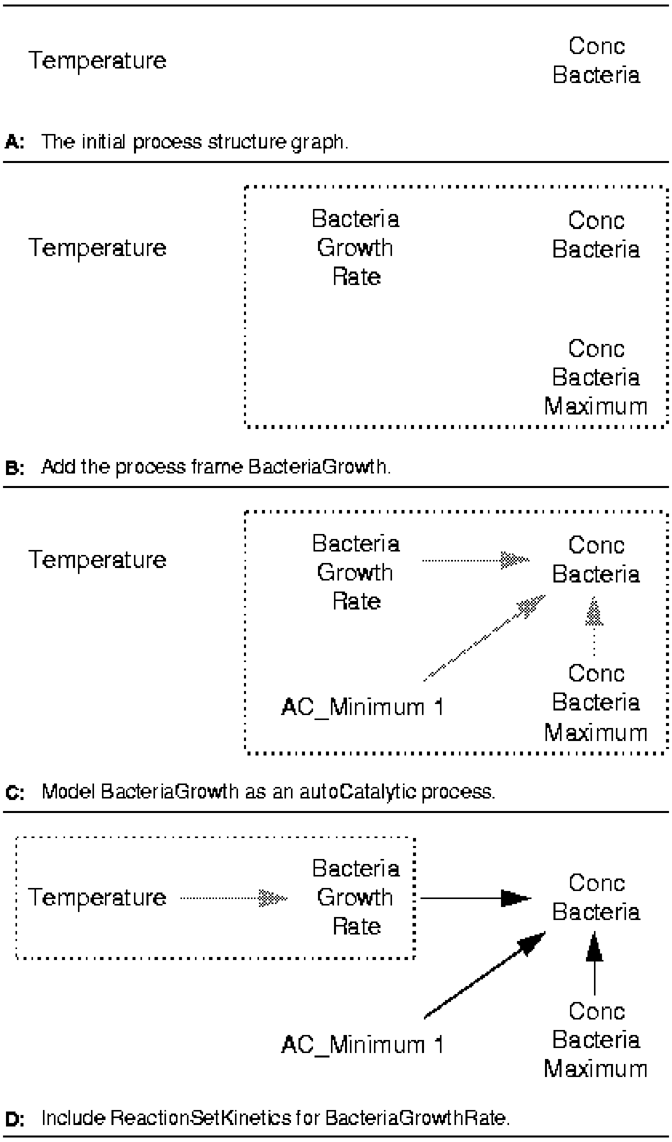


Figure 5.8: Construction of a process structure graph for the generic process of bacteria growth. Each part of the figure displays one step in the construction of the process structure graph. Boxes represent frames that are added in the step. Arrows in black represent the partial process structure graph constructed so far.

the relation between Temperature and BacteriaGrowthRate. The final process structure graph is displayed in Figure 5.8 (D).

5.4 Discussion

This chapter presented the Qualitative Process Analysis task of the Desimal approach to automated modelling of postharvest physiological phenomena. Qualitative Process Analysis uses qualitative models of generic postharvest physiological processes to construct process structure graphs for a phenomenon under study. A process structure graph is a *qualitative* model of the phenomenon under study, that represents a decomposition of the phenomenon into a set of interacting generic processes. A process structure graph is not intended for simulation of the behaviour of the phenomenon, and does therefore not contain quantitative information about the phenomenon. The next chapter describes the second subtask of the Desimal approach, called Simulation Model Construction. This subtask uses the process structure graphs to generates one or more quantitative simulation models for the phenomenon, called dynamic product models.

5.4.1 Representation of physiological processes

The knowledge graph formalism used in Desimal is based on the knowledge graph formalism described in [James, 1991]. Below the semantics of the five relation types used in Desimal are summarised, and compared to the formalism presented in [James, 1991].

- **par relations**

  A par relation relates an aggregate product property to one of the detailed product properties into which the aggregate product property can be decomposed. The par relations are also used to represent quality assignment relations between a quality attribute and the intrinsic product properties that are perceived and evaluated to assess the quality attribute. Hence, a quality attribute is treated as an aggregate product property.

  In the knowledge graph representation of [James, 1991] the par relations are used to represent that one domain concept is a part of, or an attribute of, another domain concept. In Desimal, however, the par relation type is used only to connect product properties at different levels of detail. Product properties in the DESIMAL library may have attributes. These attributes are represented as separate product properties, and may be affected by other processes. The knowledge graphs in DESIMAL only represent the behaviour of the product properties. Therefore, the influences between an attribute and a product property are represented, but the part-of relationships between an attribute and a product property are not.

- **DCAU and FCAU relations**

  In DESIMAL the CAU relation type is refined into FCAU and DCAU relation types to explicitly distinguish between functional and differential causality. A functional causal relation (FCAU) represents that one quantity affects the value of another quantity. A differential causal relation (DCAU) represents that one quantity affects the rate of change (first derivative) of another quantity.

- **AKO and EQU relations**

  In the knowledge graph formalism of [James, 1991] a frame can have a relation to any other concept in the knowledge graph. All relation types that are allowed between simple concepts are also allowed between frames.

  In DESIMAL relations at the level of frames are not allowed, all relations must be at the level of the quantities. Furthermore, only EQU and AKO relations are allowed to connect quantities in different frames. An EQU relation specifies that two quantities in different frames represent the same product property. An AKO relation specifies that a quantity in one frame is a specialisation of a quantity in another frame.

  An AKO relation is not only used to represent generalisation knowledge, but, more importantly, can also be used to represent a modelling assumption about a quantity: an AKO relation from a specific quantity to a more generic quantity then represents that the specific quantity is assumed to have the same behaviour as the generic quantity. A modelling assumption that a process is a specialisation of a more generic process is represented by AKO relations between the product properties in the frame for the specific process and the product properties in the frame for the generic process. The AKO relations specify an explicit mapping between the quantities in the specific frame and the quantities in the generic frame.

- **types of frames**
  In the knowledge graph formalism of [James, 1991] a frame can represent any complex conceptual structure consisting of several other concepts and the relations between these concepts.

  The knowledge graphs in DESIMAL use two types of frames, corresponding with the two types of building blocks of the process structure graphs. A *process frame* represents a physiological process, and may only contain DCAU and FCAU relations. A *decomposition frame* contains only PAR relations to represents the complete decomposition of one aggregate product property into detailed product properties.

The TRIPEL system [Rickel and Porter, 1994; Rickel, 1995] also uses a graph representation for the domain knowledge about plant physiology. The behavioural knowledge is represented by functional and differential influences. A functional influence between two quantities represents that the one quantity is a function of the other quantity, and thus corresponds to the FCAU relation in DESIMAL. A differential influence between two quantities represents that the rate of change of the one quantity is a function of the other quantity, and thus corresponds to the DCAU relation in DESIMAL.

The influences in TRIPEL specify the direction of change (+, −). This information is used to predict the behaviour of the variables of interest. A causal relation in the DESIMAL library does not specify a direction of change. In DESIMAL the causal relations only specify that one quantity depends on another quantity. The exact form of the dependency, including the direction of change caused by the influencing quantity, is specified in the mathematical model that will be selected for the causal relation during the Simulation Model Construction task.

The TRIPEL system uses two constructs to represent differences in the level of detail. On the one hand, *explanation* relations are used to represent that a process can in more detail be described as a set of subprocesses with intermediate quantities.[1] On the other hand, *encapsulation* relations are used to represent that a quantity can be decomposed into subquantities with intermediate processes. In our view, the explanation and encapsulation relations represent the same kind of knowledge (namely about possible decompositions), but from different viewpoints (explanations from the viewpoint of the processes and encapsulations from the viewpoint of the quantities). The decomposition frames

[1] Actually, the explanation relation is defined on the influences, but an influence can be seen as a primitive process.

in Desimal correspond to encapsulation relations.

5.4.2 Specification of the modelling question

Qualitative Process Analysis starts from a modelling question that consists of a set of quantities of interest, a set of exogenous quantities, and possibly a set of irrelevant quantities. The phenomenon has to be modelled in such a way that the quantities of interest are described explicitly. All quantities of interest have to be included in the process structure graph. The modelling task fails if it is not possible to create a connected process structure graph containing all specified quantities of interest.

This definition of quantities of interest corresponds to the definition given in [Top, 1993, page 32], where quantities of interest defined as the 'measurable qualities through which the model must mimic the observed system. Quantities of interest are the principal entities in the model'. His *evolutionary modelling* method starts from a question about a system under study. The question mentions one or more quantities of interest the behaviour of which has to be described by the model. Contrary to the approaches to automated model construction, the structure of the system under study is not formally specified. It is supposed that the modeller has a diagram or mental model of the system under study, and that the assumptions made by the modeller during the modelling process correspond with this diagram or mental model.

The model construction task in the Tripel system [Rickel and Porter, 1994; Rickel, 1995] starts from a prediction question that consists of variables of interest and driving variables. For each driving variable a value or a direction of change is specified. The goal of the modelling task is to build a model that describes the changes in the variables of interest caused by the values and directions of change of the driving variables. The model construction task selects a suitable system boundary. The variables on this boundary are called exogenous variables. These exogenous variables are thus not given by the modeller, but are determined by the Tripel system. A variable in a model constructed by Tripel is exogenous if it has a significant influence on a variable of interest, but any influence on the variable itself is not significant. A driving variable need not be an exogenous variable, but may be influenced by other variables in the constructed model.

In the Desimal approach the exogenous variables are usually external factors, such as temperature and air composition, and correspond with the driving variables in TRIPEL. As the goal of our approach is to describe the behaviour of the quantities of interest in function of the behaviour of the exogenous quantities, the modelling question does not specify initial values for the exogenous variables. As in the TRIPEL approach, an exogenous variable may be influenced by quantities within the phenomenon under study. However, influences on the exogenous quantity that do not belong to the phenomenon under study are ignored. This definition allows that the effects of a product in a confined space, *e.g.* a package, on the immediate environment of the product are included in the process structure graph, but changes in the environment caused by the package, such as diffusion through the package material, are ignored.

5.4.3 Determining the relevant system

The approaches to automated model construction reviewed in Section 4.2.2 produce the simplest sufficient model for a given query. These models are constructed for testing hypotheses about the *behaviour* of the system under study: given a set of values and derivatives for the driving variables, the only purpose of the model is to describe the changes in the quantities of interest caused by the driving variables. This justifies the automatic selection of the simplest possible model. The DESIMAL approach aims at supporting the modeller in the decomposition of the phenomenon under study into a set of generic or fundamental processes. Therefore, every quantity is considered relevant, that is directly or indirectly affected by an exogenous quantity, and that itself directly or indirectly affects a quantity of interest. The alternative process decompositions are presented to the modeller, who can then select the suitable decomposition. In this way, the modeller can test hypotheses about the appropriate *process decomposition* for the phenomenon under study.

In the DESIMAL approach first a comprehensive process structure graph is constructed that contains all influences of the exogenous quantities on the quantities of interest. The comprehensive process structure graph is constructed by backward chaining over the relations in the knowledge graph frames, and may also contain influences from quantities other than the exogenous quantities. If the phenomenon under study involves quantities that can be described at different levels of detail, then for each level of detail a separate comprehensive process structure graph is constructed. Next, each constructed comprehensive process structure graph is simplified into a process structure graph that describes the part of the phenomenon that is relevant for the modelling question. This process structure graph is determined by selecting all frames on a path between an exogenous quantity and a quantity of interest.

The relevant process structure graph represents all effects of the exogenous quantities on the quantities of interest. Usually, this process structure graph has to be further simplified. Simplification is done by the modeller, who has to adapt the modelling question, and specify some quantities in the constructed model to be irrelevant or exogenous. The Qualitative Process Analysis task has to be executed again to construct the smaller process structure graph. Hence, the simplification is done after the construction of the comprehensive process structure graph for the phenomenon under study. This supports the idea that a modeller has to know and understand the complete behaviour before certain parts of that behaviour can safely be disregarded.

Chapter 6

Simulation Model Construction

6.1 Introduction

The DESIMAL approach to automated modelling support consists of two subtasks. In the first subtask, *Qualitative Process Analysis* (see Chapter 5), qualitative models, called *Process Structure Graphs* (PSG), are constructed that represent how a phenomenon under study can be decomposed into a set of interacting generic processes.

The process structure graphs are not used for simulation, but serve as requirement specifications for the second subtask, *Simulation Model Construction*. A process structure graph specifies the processes that are occurring in a phenomenon under study and the interactions between the processes. In the Simulation Model Construction task, for each process structure graph one or more quantitative simulation models, called *Dynamic Product Models* (DPM), are constructed by selecting appropriate quantitative models for each process and for each interaction. Although a simulation model could be constructed immediately from the observed behaviour of the phenomenon, the process structure graph helps to construct simulation models that describe the underlying processes rather than only the observed behaviour of the phenomenon.

The Simulation Model Construction task uses the applicability knowledge and the mathematical knowledge in the DESIMAL library. This is displayed in Figure 6.1, which is the lower part of Figure 4.2 discussed in Section 4.3.

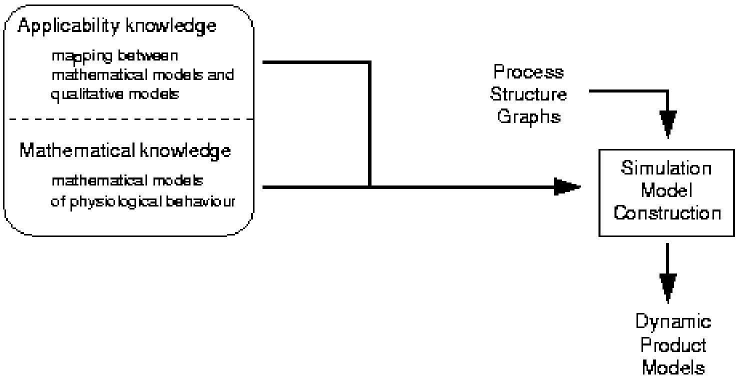

Applicability knowledge
mapping between mathematical models and qualitative models
Mathematical knowledge
mathematical models of physiological behaviour
Process Structure Graphs
Simulation Model Construction
Dynamic Product Models

Figure 6.1: Overview of the Simulation Model Construction task.

A dynamic product model is a *compositional simulation model*, consisting of a set of mathematical models and a simulation-control component to activate these models. The models in a dynamic product model are connected by interactions that transfer the value of an output variable of one model to a value for an input variable of another model.

The mathematical models are formulated as sets of differential and algebraic equations. The specification of a mathematical model in the DESIMAL library consists of its formulation, its variables, and the operating conditions for which the formulation is valid.

The dynamic product model describes the physiological behaviour underlying the quality change of an agricultural product during postharvest distribution. The quality change phenomenon may involve processes that are active only during a certain period of the distribution. To efficiently simulate the behaviour of the quality change phenomenon, the dynamic product model must be able to dynamically activate and deactivate the mathematical models for such processes during a simulation experiment. To this aim, the dynamic product model has a simulation-control component that is a declarative specification of the conditions at which the models in a dynamic product model have to be activated and deactivated. In this way, a dynamic product model describes the activation and deactivation of processes occurring in the phenomenon under study. Furthermore, the dynamic product model can activate alternative mathematical submodels for one process, depending on the actual values of the variables. In Section 6.3 the elements and the specification of a compositional simulation model are elaborated.

Each mathematical model in the DESIMAL library corresponds to one or more process frames or decomposition frames in the library. Knowledge about the application of the mathematical models to the knowledge graph frames in the DESIMAL library is represented in separate *application frames*. An application frame for a mathematical model in the DESIMAL library specifies for which knowledge graph frames the model can be applied, and specifies constraints of the mathematical model with respect to other mathematical models. The contents and the representation of the applicability knowledge in the DESIMAL library are described in Section 6.4.

The goal of the Simulation Model Construction task is to retrieve from the DESIMAL library those mathematical models that are applicable for the given process structure graph, and to construct the simulation-control component to activate these mathematical models. The Simulation Model Construction task is further described in Section 6.5. Finally, Section 6.6 describes how the specification of a dynamic product model is translated into an executable simulation model.

6.2 Example: modelling an enzymatic reaction

The Simulation Model Construction task is illustrated by the construction of an executable simulation model for an enzymatic reaction. An enzymatic reaction is a chemical reaction in which one reactant, an enzyme E, is a catalyst for the conversion of the reaction substrate S into the reaction product P. The enzyme itself is not used up by the enzymatic reaction. Typically, the following overall reaction scheme is assumed:

$$\mathrm{S} + \mathrm{E} \xrightarrow{k} \mathrm{P} + \mathrm{E}.$$

The quantitative model for an enzymatic reaction depends on the assumed type of kinetics, which can be first-order (Arrhenius) or Michaelis-Menten. Assuming first-order kinetics leads to the quantitative model:

$$\begin{aligned} \tfrac{dP}{dt} &= kSE \\ \tfrac{dS}{dt} &= -kSE \end{aligned}$$

where k is the reaction rate according to Arrhenius' law.

Often Michaelis-Menten kinetics are assumed for an enzymatic reaction [Chang, 1981]. Michaelis-Menten kinetics are more complex than first-order kinetics, in that Michaelis-Menten kinetics assume a so-called active complex AC of the substrate S and the enzyme E, from which the reaction product P is formed. The active complex is assumed to be in dynamic equilibrium with the substrate and the enzyme. This equilibrium is maintained by two reactions between the substrate and the enzyme and the active complex:

$$\mathrm{S} + \mathrm{E} \underset{k_{s2}}{\overset{k_{s1}}{\rightleftarrows}} \mathrm{AC} \xrightarrow{k_p} \mathrm{P} + \mathrm{E}.$$

Hence, assuming Michaelis-Menten kinetics leads to a complex reaction mechanism, that consists of three reactions with separate reaction rates. The complete quantitative model for an enzymatic reaction with Michaelis-Menten kinetics is:

$$\begin{aligned} V_{max} &= k_p E_0 \\ K_m &= (k_{s2} + k_p)/k_{s1} \\ P &= P_0 + S_0 - S \\ \frac{dS}{dt} &= -\frac{V_{max} S}{K_m + S} \end{aligned}$$

In the remainder of this chapter, the construction of an executable simulation model for an enzymatic reaction is used as a running example to illustrate the task and knowledge representation for Simulation Model Construction.

6.3 Compositionality in simulation models

The dynamic product models constructed in DESIMAL are compositional simulation models. In [van Langevelde *et al.*, 1992], a ***compositional system*** is defined to consist of a set of components, a set of interactions between the components, and a control level. The control level is a declarative specification of the conditions at which the components must be activated and deactivated. Interactions between components are used to transfer the values of output variables of one component to values for input variables of another component. Each component represents an aspect of the modelled system or phenomenon that can be described in isolation.

The dynamic product models constructed in DESIMAL are compositional systems in the above sense, and consist of mathematical models, interactions between these models, and one simulation-control component.

6.3.1 Mathematical models

The mathematical models contain the mathematical equations of a dynamic product model. Each mathematical model describes the changes in intrinsic product properties caused by one process or one aspect of a process.

Figure 6.2 displays the mathematical models for an enzymatic reaction with Michaelis-Menten kinetics. The three models in the left column describe the changes in P and S with Michaelis-Menten kinetics. The model Michaelis-Menten is the generic model, and has no restrictions on its operating region. The models linear-MM and exponential-MM are approximations of this model, and are only valid in well-defined operating regions. The input variables Vmax and Km of these three models are combinations of other variables. The calculation of Vmax and Km is specified in separate models maximum-rate-MM and specificity-MM. The variable Vmax is a function of the concentration of active enzyme E and the chemical rate kp. The variable Km is a function of chemical rates ks1, ks2 and kp of the individual chemical reactions in the enzymatic conversion.

The specification of a mathematical model consists of its formulation, its operating conditions, and the variables that appear in the formulation and in the operating conditions.

- **variables**
  This section specifies the variables that appear in the formulation and in the operating conditions of the mathematical model. The variables are listed by their role in the mathematical model: input, internal or output. A variable that is an input of a model may also be output of the model. An example is the variable S in the models linear-MM, exponential-MM, and Michaelis-Menten in Figure 6.2.

- **formulation**
  Each formula in the formulation of a mathematical model has to be an assignment of the value of the right hand side expression of the formula to an internal or output variable of the model. Implicit equations, such as $x + y = 0$, are not allowed.

  An expression integrate(diffeq) in a formula specifies that the value of the output variable is the result of an integration of the differential equation diffeq over some small time step. The time step is determined by the numerical integration routine applied when executing the dynamic product model, and is not part of the specification of the mathematical model, nor of the dynamic product model. The integrate expression in the formulation of the model linear-MM specifies the differential equation $dS/dt = -V_{max}$.

- **operating conditions**
  The operating conditions specify the value ranges of the variables in the dynamic product model for which the formulation of the model is valid. The operating conditions are specified as (in)equalities between variables used in the dynamic product model. Hence, in the operating conditions for one mathematical model also variables from other mathematical models may be used. An example is the Km variable in the model linear-MM. Although Km is not used in the formulation of the model, the value of Km determines when it is allowed to approximate the rates of change of S and P by a linear function.

An important property for a library of models is that one model must not make assumptions about other models in the library, because this would imply that once that model is applied, then also the other models have to be applied. The models are not independent any more. This is called the No Function In Structure principle [de Kleer and Brown, 1984]. If a variable in the operating condition of a mathematical model in the DESIMAL library is not used in the model formulation, then this means that the model in fact has an additional input variable. No assumption is made about which model has to be used to calculate the value of the variable. Hence, the No Function In Structure principle is not violated.

6.3.2 Interactions between models

In general, when a simulation model is developed by reusing previously developed models, the value of the output variable of one model has to be transformed before it can be used as an input for another model. These transformations are often hidden in the formulations of the models.

In a dynamic product models constructed by DESIMAL these transformations are explicitly specified by interactions. An *interaction* connects one output of a mathematical model to one input of another mathematical model, and specifies a conversion function to transfer the value of the output variable to a

```
model linear-MM
  variables
    input S, Vmax, Km
    output S, P
  operating conditions
    $S > 2 \cdot Km$
  formulation
    $P := P(0) + S(0) - S$
    $S :=$ integrate($-$Vmax)
end model

model exponential-MM
  variables
    input S, Vmax, Km
    output S, P
  operating conditions
    $S < Km/2$
  formulation
    $P := P(0) + S(0) - S$
    $S :=$ integrate( $-(Vmax/Km) \cdot S$ )
end model

model Michaelis-Menten
  variables
    input S, Vmax, Km
    output S, P
  formulation
    $P := P(0) + S(0) - S$
    $S :=$ integrate( $-(Vmax \cdot S)/(Km+S)$ )
end model

model maximum-rate-MM
  variables
    input kp, E
    output Vmax
  formulation
    $Vmax := kp \cdot E$
end model

model specificity-MM
  variables
    input ks1, ks2, kp
    output Km
  formulation
    $Km := (ks2+kp)/ks1$
end model
```

Figure 6.2: Mathematical models for enzymatic reactions assuming Michaelis-Menten kinetics.

value of the input variable. Note that with this definition, several interactions are necessary when two models share more than one variable.

An interaction is not intended to describe a part of the physiological behaviour of the phenomenon under study, but is a technical construct to accommodate the composition of previously developed mathematical models. Therefore, the conversion function has to be an algebraic and time-independent function, in which only the value of the variable to be transformed and constants may be used.

Besides the conversion function, the specification of an interaction contains one source variable, one target variable and the quantities represented by these variables. The variables in the models need not be unique in a dynamic product model. However, each quantity in the qualitative level of the DESIMAL library has a unique name and a well-defined semantics. Hence, the interaction is in fact defined in terms of the quantities. For example, the interaction in Figure 6.3 specifies the conversion of the value of any variable representing the quantity Temperature to a value for any variable representing the quantity AbsoluteTemperature. This interaction can be applied to convert the observed temperature in the product environment into an input variable for the model Arrhenius, that describes the dependence of the rate of a chemical reaction on temperature.

```
interaction Centigrade-to-Kelvin
    variables
        source T
        target Tabs
    quantities
        source Temperature
        target AbsoluteTemperature
    conversion function
        Tabs := T + 273.15
end interaction
```

Figure 6.3: The specification of an interaction to transfer temperatures on the Celsius scale to absolute temperatures.

Often, an output variable of a model need not be converted before it can be used as an input of another model. In these cases, the variables in the two models are connected by an interaction in which the conversion function is identity. The interaction then merely indicates that the output of one model is used as an input of another model.

6.3.3 The simulation-control component

A dynamic product model has a separate simulation-control component, that controls the execution of the dynamic product model, and is active throughout a simulation experiment. The simulation-control component is a declarative specification of the conditions at which the models in the dynamic product model have to be activated and deactivated.

Three requirements have to be satisfied before a mathematical model in the dynamic product model can be active at a certain time point: (1) the process described by the model must be occurring at that time point, (2) the values of all input variables of the model must be known, and (3) the operating conditions of the model must be satisfied. The simulation-control component contains rules for each of these requirements. These rules are processed separately and continuously during a simulation experiment.

A simulation-control component consists of three sections: Initialisation, Execution, and Termination. The sections are described below.

Initialisation This section specifies the initial values for variables in the dynamic product model, and may also activate models to calculate variables that depend only on variables that have a constant value throughout the simulation experiment. These models are thus activated only once, and must be time-independent. The Initialisation section is processed at the start of the simulation experiment.

Execution The dynamic product model is a continuous time simulation model, and uses a numerical integrator to run the simulation, see Section 6.6. The numerical integrator divides the continuous simulation interval into a number of small time steps. At the start of each time step the rules in the Execution section are processed to determine which models in the dynamic product model have to be activated or deactivated to simulate the behaviour during that time step. Figure 6.4 shows the execution rules of a simulation-control component in a dynamic product model that contains the models for an enzymatic reactions displayed in Figure 6.2 and two identity interactions transfer-Vmax and transfer-Km that transfer the variables Vmax and Km between the models. The Execution section contains three types of rules:

```
Execution
 1.  if     Substrate > 0 and ActiveEnzyme > 0
     then   occurring-process( EnzymaticReaction )
 2.  if     occurring-process( EnzymaticReaction )
     then   applicable-model( Michaelis-Menten )
 3.  if     occurring-process( EnzymaticReaction )
      and   Substrate > 2*Km
     then   applicable-model( linear-MM )
 4.  if     occurring-process( EnzymaticReaction )
      and   Substrate < Km/2
     then   applicable-model( exponential-MM )
 5.  if     occurring-process( EnzymaticReaction )
     then   applicable-model( maximum-rate-MM )
 6.  if     occurring-process( EnzymaticReaction )
     then   applicable-model( specificity-MM )
 7.  if     applicable-model( maximum-rate-MM )
     then   model-activation( maximum-rate-MM )
 8.  if     applicable-model( specificity-MM )
     then   model-activation( specificity-MM )
 9.  if     applicable-model( linear-MM )
      and   model-terminated( maximum-rate-MM )
      and   model-terminated( specificity-MM )
     then   interaction-activation( transfer-Vmax)
      and   interaction-activation( transfer-Km)
      and   model-activation( linear-MM )
10.  if     applicable-model( exponential-MM )
      and   model-terminated( maximum-rate-MM )
      and   model-terminated( specificity-MM )
     then   interaction-activation( transfer-Vmax)
      and   interaction-activation( transfer-Km)
      and   model-activation( exponential-MM )
11.  if     applicable-model( Michaelis-Menten )
      and   not applicable-model( linear-MM )
      and   not applicable-model( exponential-MM )
      and   model-terminated( maximum-rate-MM )
      and   model-terminated( specificity-MM )
     then   interaction-activation( transfer-Vmax)
      and   interaction-activation( transfer-Km)
      and   model-activation( exponential-MM )
```

Figure 6.4: The Execution section in the simulation-control component of a dynamic product model for an enzymatic reaction with Michaelis-Menten kinetics.

- occurring-process rules

  The first rules are occurring-process rules. The simulation-control component must have an occurring-process rule for each process that is modelled in the dynamic product model. This rule type specifies the conditions at which the process will occur. An occurring-process rule is expressed in terms of the quantities that are involved in the process.

  For example, rule 1. in the Execution section in Figure 6.4 specifies that the EnzymaticReaction occurs as long as there is a substrate and an active enzyme.

  These rules are specified for process frames only. Decomposition frames specify a relationship between quantities, that is similar to the interactions between models. Therefore, decomposition frames are activated in the same way as interactions. This will be described below.

- applicable-model rules

  The occurring-process rules specify which processes are active. The models in the DPM that are applicable to simulate the behaviour of these processes are specified by applicable-model rules.

  The simulation-control component must have an applicable-model rule for each model in the dynamic product model that describes a process. Again, models for decomposition frames are treated differently. This rule type specifies the (combinations of) processes for which the model can be applied, and the operating conditions for the model, as specified in the DESIMAL library.

  Rules 2. thru 6. in Figure 6.4 are the applicable-model rules. Of these, rule 3. specifies that the model linear-MM can be applied to describe the process frame EnzymaticReaction, if the process is occurring and if the value of the variable representing Substrate is at least twice as large as the value of the variable that represents Km.

- model-activation rules

  The applicable-model rules specify which models are applicable for the occurring processes. A model can be activated if it is applicable for an occurring process, and if the values of the input variables of the model are known. The latter condition ensures that the applicable models are activated in the order of computation.

  The simulation-control component must have a model-activation rule for each model in the dynamic product model. The condition part of this rule type is a logical combination of applicable-model and model-terminated clauses. The action part is a sequence of interaction-activation clauses followed by one model-activation clause.

The applicable-model clauses specify which models have to be applicable and which models may not be applicable, in order to activate the current model. If several models in the dynamic product model are applicable for a process, then the applicable-model rules for these models may specify overlapping conditions. The applicable-model clauses in the model-activation rules for these models must be formulated such, that always exactly one of these models is activated. For example, three models are available to describe the changes in the Substrate. The models linear-MM and exponential-MM have disjunct operating regions, but the model Michaelis-Menten is always applicable. The last three rules in Figure 6.4 specify that the approximations must be activated whenever these models are applicable, and that the model Michaelis-Menten must be activated only if the approximation models are not applicable.

The model-terminated clauses ensure that the model is activated, only if other models that calculate the input variables have been activated and terminated during the current time step. The model-terminated clauses explicitly represent the order of computation of the applicable models.

Before a model can be activated, the values for its input variables have to be transferred from the models that calculated these values. If the input variable of the model to be activated represent the same quantity as the output of another model, then an interaction has to be used to link the input variable to the output variable, If the input variable represents a quantity that is a composition of other quantities (as represented by a decomposition frame in the process structure graph), then the corresponding model for this decomposition frame has to be used to calculate the new value for the input variable. Note that the decomposition models are treated as interactions, with several source variables an one target variable. The necessary interactions are specified by interaction-activation clauses.

At the start of each time step, the rules in the Execution section are processed to decide which models in the dynamic product model have to be activated. These models are activated to calculate new values of the output variables. During one time step, a model can be activated only once. Hence, loops in the dynamic product model will always involve two time steps.

Termination This section contains a simulation-termination condition that specifies when the dynamic product model normally has to terminate. The dynamic product model terminates successfully, if this condition is satisfied. The dynamic product model terminates unsuccessfully if at a certain time point no models can be activated. In that case, the simulation has reached a state that was not foreseen during model construction, so the dynamic product model was used outside its region of operation.

The Termination section in the simulation-control component of a dynamic product model for an enzymatic reaction may be as follows:

```
Termination
    if     ActiveEnzyme = 0 or Substrate = 0
    then   simulation-termination
```

6.4 Applicability documentation

So far, two representation formalisms have been described for the postharvest physiological behaviour of agricultural products: a qualitative representation in terms of processes and decompositions of quantities (Section 5.2.3), and a quantitative representation in terms of mathematical equations (Section 6.3).

To connect these two representation formalisms, the DESIMAL library has an intermediate knowledge level consisting of *application frames*. An application frame relates one mathematical model to one or more qualitative knowledge graph frames. The application frames provide additional knowledge that is not present in the specifications of the mathematical models, but that is necessary for automated construction of the dynamic product models.

An application frame for a mathematical model specifies a *process structure* consisting of the qualitative knowledge graph frames for which the mathematical model is applicable, and an *interpretation* that maps the variables of the mathematical model onto the quantities in the process structure.

6.4.1 The process-structure specification

The process-structure specifies the knowledge graph for which the mathematical model is applicable, and is a conjunction of four predicates:

- A term describes-process(pf) specifies that the model describes the process frame pf.
- A term describes-decomposition(df) specifies that the model describes the decomposition frame df.
- A term ignores-process(pf) specifies that the mathematical model is formulated with the assumption that the quantities in the described processes and decompositions are not influenced by the process frame pf.
- A term ignores-decomposition(df) specifies that one of the quantities in the process structure has to be described as an aggregate quantity, and cannot be decomposed into its subquantities, as described in the decomposition frame df.

A term ignores-decomposition(df) can only be used to specify that a quantity in the process structure graph has to be described as an aggregate quantity. This quantity may not be an input of the model, because this would represent an assumption about the way in which the inputs of the model are calculated. This would violate the aforementioned No Function In Structure principle.

A term ignores-decomposition(df) cannot be used to specify that one quantity in the process structure graph may not be a subquantity of another quantity in the process structure graph. This would mean that the behaviour of the quantity depends on its use, which is not allowed.

A conjunction of the predicates specifies that the model describes a connected graph of the specified frames. For example, the process-structure of an application frame for a model describing the effect of bacteria on vessel blocking would be:

```
describes-decomposition(Dec-BlockedVessels) and
describes-process(BacteriaBlocking) and
ignores-process(AirBlocking).
```

This model disregards the vessel blocking caused by air bubbles that entered the flower stem during a period of dry storage. However, it still contains the quantity AirBlockedVessels, which will therefore be treated as an input of the model.

Disjunctions of the predicates are not allowed, because an application frame specifies how *one* mathematical model can be applied to *one* knowledge graph. Consequently, when a mathematical model is applicable to different knowledge graphs, then separate application frames have to be specified for each knowledge graph.

6.4.2 The interpretation of the model variables

The interpretation specifies how variables of the mathematical model map to quantities in the knowledge graph specified in the process-structure. The mapping need not be complete: variables of the mathematical model may represent quantities not included in the knowledge graph frames for which the model is applicable (e.g. Vmax and Km in the models displayed in Figure 6.2), and quantities in these knowledge graph frames may be left implicit in the model formulation. The interpretation is specified by the following terms:

A term mapping(v,q) specifies that the model variable v represents the quantity q. If this quantity does not occur in the knowledge graph frames for which the model is applicable, then v can be an internal variable of the model, or v can be an additional input variable of the model. A model cannot have output variables that represent other quantities than those in the knowledge graph frames for which the model is applicable, as this would imply that the model describes more processes than specified by the process-structure.

A term implicit(q) specifies that the quantity q is described implicitly in the formulation of the model. This term specifies that, when this model is applied for the specified knowledge graph, then the quantity q appearing in a frame in the knowledge graph, is covered by this model.

6.4.3 Examples

Figure 6.5 shows the application frames for the models linear-MM and maximum-rate-MM in Figure 6.2. The first application frame specifies that the model linear-MM describes the behaviour of the quantity Substrate in the process frame EnzymaticReaction. The term implicit(ApparentRate) specifies that the apparent rate of the enzymatic reaction is included in the model formulation, but is not described explicitly. Instead the apparent rate is a combination of two quantities MM_MaximumRate and MM_Specificity, represented by the variables Vmax and Km. These variables are additional inputs to the model, that are either input to the dynamic product model, or calculated by other models. According to the second application frame the quantity MM_MaximumRate can be calculated by the model maximum-rate-MM.

```
application frame linear-MM
  process-structure
    describes-process(EnzymaticReaction)
  interpretation
    implicit(ApparentRate)
    mapping(S, Substrate)
    mapping(P, Product)
    mapping(Vmax, MM_MaximumRate)
    mapping(Km, MM_SaturationRate)
end application frame

application frame maximum-rate-MM
  process-structure
    describes-process(EnzymaticReaction)
  interpretation
    mapping(kp, EnzymaticReactionRate)
    mapping(E, ActiveEnzyme)
    mapping(Vmax, MM_MaximumRate)
end application frame
```

Figure 6.5: Application frames for two models for the process frame EnzymaticReaction.

6.5 The simulation model construction task

The Simulation Model Construction task starts from a process structure graph constructed in Qualitative Process Analysis, and constructs one or more dynamic product models for this process structure graph. Which process structure graph is used, is a decision of the modeller, who has to select the process structure graph that represents the preferred decomposition of the phenomenon under study.

As described in Section 4.3, the Simulation Model Construction task involves reasoning about the reuse of the mathematical models in the Desimal library. The result of the Simulation Model Construction task is a specification of a dynamic product model. Such a specification consists of a set of mathematical models that are applicable for the processes in the process structure graph, and a simulation-control component, that specifies in which order the models have to be activated.

The Simulation Model Construction task consists of two subtasks. The first task involves the selection of all mathematical models from the Desimal library that are applicable for the processes represented in the process structure graph. The second subtask evaluates the applicable models, and constructs one or more *consistent simulation models*. A consistent simulation model describes the processes that are occurring at a certain point during the simulation. A consistent model has exactly one mathematical model for each relation in the process structure subgraph for the occurring processes. The consistent simulation models in a dynamic product model are specified by model-activation rules in the simulation-control component.

The simulation-control component is constructed as part of the subtasks of the Simulation Model Construction task. First, occurring-process rules are generated for the processes represented in the process structure graph. The occurring-process rules are generated after the modeller has selected a process structure graph, from the process frames in the process structure graph and their occurrence conditions.

After the applicable models have been retrieved, for each applicable model one applicable-model rule is generated, that specifies the process in the process structure for which the model is applicable and the operating conditions of the model.

After the consistent simulation models have been identified, the model-activation rules are generated. A model-activation rule for a model specifies which models and interactions have to be activated and terminated, before that model can be activated. One applicable model may have several such rules, corresponding to different consistent simulation models in which the model is used.

6.5.1 Retrieval of applicable models

The first subtask of Simulation Model Construction is the retrieval of the applicable models. This subtask selects from the mathematical models in the Desimal library an *applicable model set*, containing all models that describe a part of the behaviour represented in the process structure graph. The dynamic product model will contain the complete set of applicable models.

To retrieve the applicable models, the process-structure conditions in the application frames in the Desimal library are used. A mathematical model is applicable for a subgraph of the process structure graph, if the following conditions hold for the subgraph:

- the subgraph must be connected and must contain all process frames and decomposition frames in the process-structure condition of the application frame for the mathematical model;
- the subgraph must not contain additional process frames or decomposition frames;
- the quantities in the subgraph must not be influenced by excluded process frames, that are specified by ignores-process and ignores-decomposition terms in the process-structure condition of the application frame for the mathematical model.

These requirements express that the process structure graph must be divided into subgraphs in such a way that each subgraph exactly corresponds to one or more mathematical models in the DESIMAL library. All frames in the process structure graph are considered to be relevant, and therefore have to be modelled. Furthermore, models describing behaviour that is not represented in the process structure graph, cannot be applicable.

The requirements are checked with a simple graph matching algorithm. The algorithm stops when applicable models have been found for all relations in the process structure graph, and when all mathematical models in the DESIMAL library have been checked.

The exact mapping between a subgraph in the process structure graph and an applicable mathematical model is specified in the interpretation section of the application frame for the mathematical model. The interpretation section specifies which quantities in the subgraph are represented by variables in the mathematical model, and which quantities are left implicit in the formulation of the mathematical model. The relations between these quantities are supposed to be covered by the formulation of the mathematical model. More precisely, if a mathematical model is applicable for a process frame, and the model has two variables that are mapped to quantities in the process frame, then it is supposed that the formulation of the mathematical model describes all knowledge graph relations between these quantities that are included in the subgraph of the process structure graph for which the mathematical model is applicable.

The mathematical model may not cover all quantities in the process structure subgraph. In that case additional models are retrieved for the qualitative relations that involve one or more quantities that are not covered. Furthermore, if the mathematical model has variables representing quantities that are not included in the subgraph for which the mathematical model is applicable, then additional mathematical models are retrieved that describe the changes in these quantities in terms of quantities included in the process structure subgraph. In the end, a set of mathematical models must be selected, so that all quantities in the process structure subgraph are covered by one of these models.

6.5.2 Selection of consistent simulation models

The *applicable model set* contains all mathematical models that describe one or more relations in the process structure graph. These models are all included in the dynamic product model. The applicable model set may contain alternative models for a relation in the process structure graph.

The specification of the model-activation rules has to be such that at each time point during a simulation experiment exactly one mathematical model in the dynamic product model is activated for each relation in the process structure subgraph that represents the processes that are occurring at that time point.

The set of active models for the occurring processes is called a consistent simulation model. The goal of the second subtask of Simulation Model Construction is to select such consistent simulation models from the applicable model set for this subgraph.

If each relation in the process structure graph is covered by exactly one applicable model, then the applicable model set forms one consistent simulation model. When several models are applicable for a relation in the process structure graph, then also several consistent simulation models are found.

The set of consistent simulation models is determined by the algorithm in Figure 6.6.

6.5.2.1 Example

The model-activation rules in the simulation-control component in Figure 6.4 specifies three consistent simulation models inside the dynamic product model for an enzymatic reaction. These consistent simulation models only differ in the mathematical model that is used for the qualitative relation between the quantities ActiveEnzyme and Substrate in the process frame EnzymaticReaction. The consistent simulation models are specified by the model-activation rules. For the models maximum-rate-MM and specificity-MM only one model-

```
foreach relation r in the process structure subgraph for the occurring processes:
   foreach model m that is applicable the relation
      – create a new consistent simulation model, containing the models
        already selected for other relations in the subgraph.
      – add the selected model to the consistent simulation model.
      – mark the relations in the subgraph between quantities represented
        by the variables of selected model as covered.
   end foreach
end foreach
```

Figure 6.6: The algorithm to select the consistent simulation models.

activation rule is specified, so that these models are always used when they are applicable.

6.6 Translation into an executable model

The DESIMAL system translates the specification of a dynamic product model into a PROSIM simulation model. PROSIM is a modular simulation environment for simulation of parallel processes (Sierenberg & de Gans). The implementation of a dynamic product model in PROSIM consists of the required modules define and main, and a number of process modules. Each process module implements one physiological process in the dynamic product model. The main module is active throughout a simulation experiment. The other modules can be activated and deactivated.

In the define module all variables used in the simulation model are declared. The main module of the PROSIM model implements the Initialisation and Termination sections. Figure 6.7 shows the main module for a dynamic product model with one enzymatic reaction. The contents of a main module is basically the same for all dynamic product models. First, the process modules are activated. The process modules will be executed as parallel simulation processes, each having its private control about which models to be activated. Next, the main module will wait until the simulation-termination condition is satisfied. Finally, the cancel all statement will stop the process modules, and stop the simulation experiment.

Each process in the modelled phenomenon is implemented as a subsystem, having its private set of submodels and a private simulation-control component

```
mod main
    activate Enzymatic_Reaction
    wait until (E = 0 or R = 0)
    cancel all
end mod
```

Figure 6.7: Part of the main module in a PROSIM model for the enzymatic reaction process.

to determine which of these submodels has to be activated. These subsystems can be seen as reusable generic models for the processes. This means that the dynamic product model is implemented as a hierarchical system. Instead of using one set of models with one simulation-control component that determines both which processes are occurring and which models are activated for these processes, the rules in the simulation-control component are distributed over the main module of the PROSIM model and the process modules.

Figure 6.8 shows a part of a PROSIM module for the process EnzymaticReaction. The module starts with a wait until statement containing the condition part of the occurring-process rule for the process. When this condition is satisfied, the model is activated of which the operating condition is satisfied. Each model is implemented as a macro, that changes the equations for the output variables S and P. Each model is executed (integrated) as long as the operating condition for that model hold. When the operating condition does not hold anymore, it is checked whether the process is still occurring. If so, one of the other models is activated. Otherwise, the module waits until the occurring-process condition is again satisfied, or until the module is cancelled because of a cancel all command.

6.7 Discussion

In this chapter we presented the Simulation Model Construction task, and described the levels in the DESIMAL library containing the mathematical models and the application frames, that are used in this task. We described the subtasks of Simulation Model Construction.

To conclude this chapter, some technical details of these levels are looked into, and the structure of the dynamic product models is compared with the structure of simulation models constructed by other approaches to automated

```
mod Enzymatic_Reaction
    start_EnzymaticReaction:

    wait until (S > 0 and E > 0)

    if (S > 2*Km)
        call LinearApproxMM
        integrate while (S > 2*Km)
        repeat from start_EnzymaticReaction
    end

    if (S < Km/2)
        call ExpoApproxMM
        integrate while (S < Km/2)
        repeat from start_EnzymaticReaction
    end

    if ((S >= Km/2) and (S <= 2*Km))
        call MichaelisMenten
        integrate while ((S > Km/2) and (S < 2*Km))
        repeat from start_EnzymaticReaction
    end
end mod
```

Figure 6.8: Example of a Prosim module for the enzymatic reaction process that uses the approximate submodels when possible.

modelling.

6.7.1 Specification of the mathematical models

The mathematical models in the Desimal library are formulated as assignments to output or internal variables of the mathematical model (e.g. $y = -x$). Several automated modelling approaches use a model library in which the models are formulated as equations (e.g. $x + y = 0$). The simulation systems associated with those libraries use dependency relationships between the variables to transform the equations into assignments to internal and output variables. For example, one model in the library containing the equation $x + y = 0$ can be transformed into two variable assignments to be used in the simulation model: $x = -y$ when x is the dependent variable, or into $y = -x$ when y is the dependent variable. Specifying the formulation as an equation set rather than a set of assignments has the advantage that the model library can contain one model formulation for different input-output relations between the variables. If the formulation has to be in assignment form, two separate models have to be stored in the library. As the mathematical models for postharvest physiological behaviour are in general applied with one fixed input-output relationship between the variables, an assignment form of the model formulation is sufficient. However, this does not hold for models describing physical phenomena such as diffusion and osmosis, which are important transport processes in agricultural products. For such processes the DESIMAL library has to contain several mathematical models, one for each input-output relationship between the model variables.

The variables in the mathematical models in the executable dynamic product models are time series of simulated values, so that the construct var(t) can be used to refer to the value of a variable var at time t. However, explicitly referring to the value of a variable at a specific time point is not encouraged, because the numerical integration routine that is applied in the executable dynamic product model uses a variable step size, and therefore may not calculate a value for the variable at the specified time point. However, this construct can safely be used to refer to initial values of variables (at time $t = 0$), as is done for the variables P and S in the formulations of the models linear-MM, exponential-MM and Michaelis-Menten (Figure 6.2).

6.7.2 Explicit representation of applicability knowledge

The application frames in the DESIMAL library are specified by a process-structure and an interpretation. Each application frame contains applicability knowledge for one mathematical model. The process-structure specifies the knowledge graph for which the mathematical model is applicable. The interpretation specifies a mapping between the variables in the mathematical model and the quantities in the process structure.

In [Sloof, 1998] the application frames contained also a model-structure, that specified relations between the mathematical model and other models in the DESIMAL library. The model-structure was specified by req-model and not-model terms. A term req-model(m) specified that the mathematical model can only be applied in combination with another mathematical model m. This term could imply that a quantity could have different definitions, depending on the model in which the quantity is used. This is not allowed any more. Each quantity in the DESIMAL library has a unique definition, so that whenever the quantity is used, it has to be calculated by the same set of mathematical models. Hence, the information that an output quantity of one model is an input of another model, can be derived from the interpretation in the application frame. A term not-model(m) specified that the mathematical model of the application frame and the model m are based on different modelling assumptions. Therefore, the models cannot be used both in one dynamic product model to describe one process. In many cases, models that are based on different modelling assumptions also use different quantities, so that the appropriate model can be derived from the process structure graph and the interpretation in the application frame. However, still several models may be applicable for one process structure with overlapping operating conditions. In that case, the modeller has to choose which model to use. As an example, the simulation-control component displayed in Figure 6.4 results from the decision of the modeller that the approximations of the Michaelis-Menten have to be used when possible.

6.7.3 Explicit simulation-control

A dynamic product model describes the dynamic behaviour of an agricultural product during postharvest distribution. To cover all possible conditions encountered during postharvest distribution, a dynamic product model may have several submodels with different operating regions for one process. A dynamic product model has an explicit simulation-control component to control the activation of the mathematical models. In this way, during a simulation experiment not only may different models be used, but also the set of occurring processes may change.

The simulation models constructed by the method of Iwasaki and Levy may contain several models to describe the behaviour of one quantity. The models for one quantity have different operating regions, but must be based on the same underlying assumptions. Contrary to the DESIMAL approach, the set of occurring processes cannot be changed during a simulation experiment. The simulation models do not have an explicit simulation control level. Which models will calculate the next state in the simulation experiment is determined by the values of the variables and the operating conditions of the models.

The simulation models constructed in the evolutionary modelling approach [Top, 1993] have a static model structure. If another model has to be selected for an aspect of the system under study, a new simulation model has to be constructed, by changing the functional decomposition, by changing one or more assumptions about the physical processes, or by changing assumptions about the mathematical relations for the physical processes.

6.7.4 The executable dynamic product models

The executable dynamic product models are generated in the PROSIM simulation language. The structure of a PROSIM model slightly differs from the structure in the formal specification of a dynamic product model.

An executable dynamic product model in PROSIM consists of a main module and several process modules. The main module starts and stops the dynamic product model, and activates the process modules for all processes modelled in the dynamic product model. The rules that determine when the processes occur and which models are activated to simulate the processes, are distributed among the process modules. A process module in an executable dynamic product model can be seen as a small compositional component within the compositional structure of the dynamic product model. The control level of a process module consists of the applicable-model and model-activation rules.

The process modules may only contain mathematical relations, which are primitive components. Hence, the executable dynamic product model is restricted to three hierarchical levels. The highest level is the main module, the middle level contains the process modules, and the lowest level contains the mathematical relations (implemented as macros that are called from the process modules). These three levels correspond to the three types of rules in the specification of the simulation-control component. For this reason, the executable dynamic product model is treated as a non-hierarchical compositional system, corresponding to the non-hierarchical simulation-control component in the specification of a dynamic product model.

Chapter 7

The DESIMAL system

7.1 Introduction

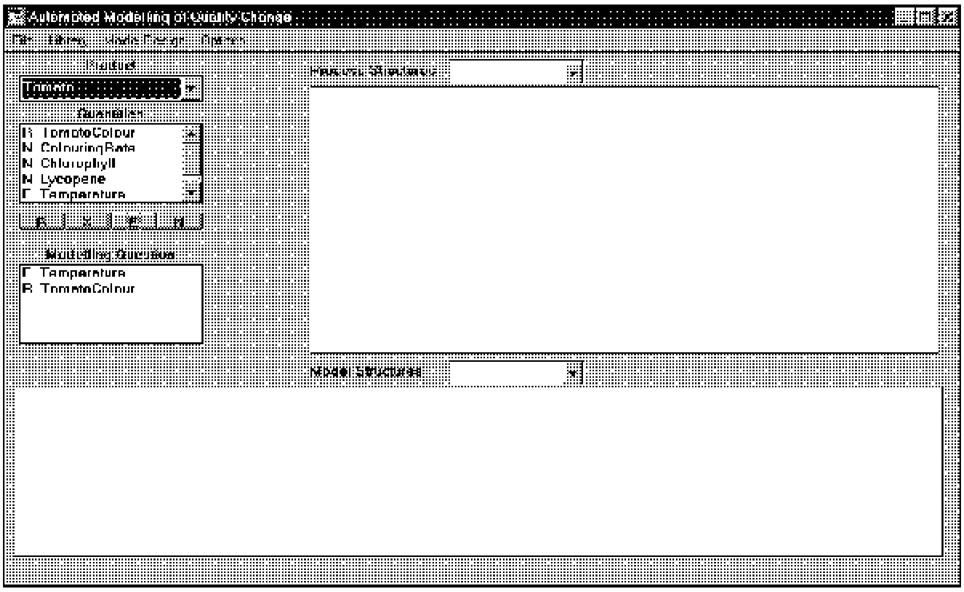
The automated modelling method described in the previous chapters has been implemented in a software system, called DESIMAL, which is an acronym for 'Design and Specification of Interacting MAthematicaL models'.

In this chapter, the DESIMAL system is described. The next section illustrates the functionality of the DESIMAL system by showing how the modeller is supported in the construction of a model for a phenomenon under study. Section 7.3 discusses the implementation of the DESIMAL system. Section 7.4 describes the generation of executable PROSIM models from the formal specifications of the dynamic product models. In Section 7.5 the quantities and the frames in the DESIMAL library are defined. Finally, in Section 7.6 a number of modelling cases are elaborated.

7.2 Functionality

This section illustrates how the DESIMAL system supports the construction of an executable quality change model for the colour development in tomatoes as a function of a temperature scenario. It is shown which actions are performed by the modeller, and which actions are performed by the DESIMAL system.

First the modeller must provide a modelling question about the phenomenon



Tomato

R TomatoColour
N ColouringRate
N Chlorophyll
N Lycopene
F Temperature

F Temperature
R TomatoColour

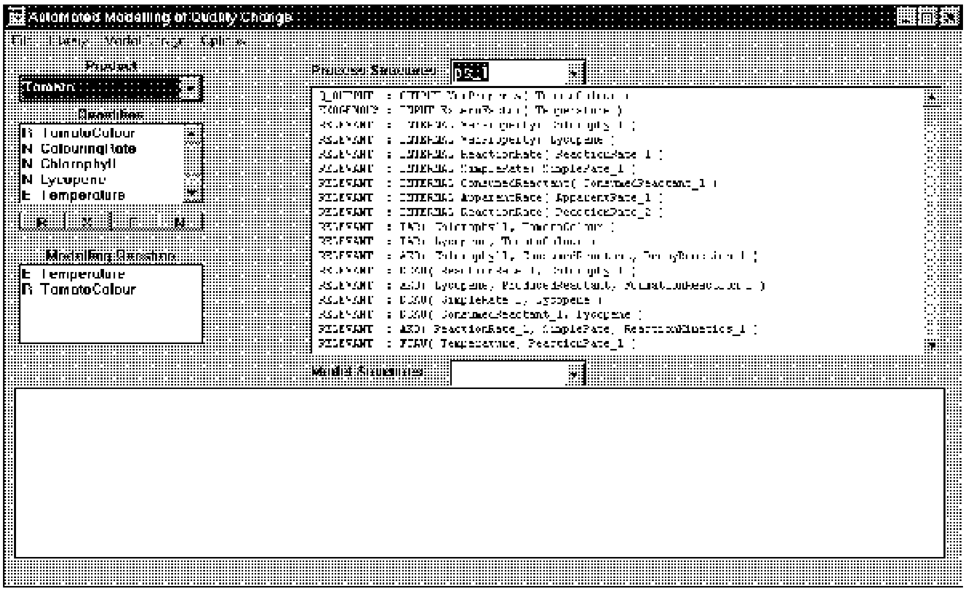
Figure 7.1: The main screen of the application.

under study. This is shown in Figure 7.1. The modeller starts by selecting the product in which the phenomenon is studied. The Desimal system then displays the quantities that are relevant for that product. These quantities include the intrinsic properties of the selected product, external factors, and possibly quality attributes of the selected product. The modelling question is specified by selecting a quantity from the list, and subsequently pressing one of the buttons below the list of quantities. By pressing the button R, the quantity is added to the modelling question as a required quantity: the constructed model is valid if it describes the behaviour of this quantity. The button X adds the quantity as an irrelevant quantity: the constructed model is valid if it does not include the quantity. By pressing the button E the selected quantity is included as an exogenous quantity of the model to be constructed. The fourth button, N, removes a quantity from the modelling question. The modelling question must contain at least one required quantity and one exogenous quantity.

The next step is to construct the process structure graphs for the modelling question. The process structure graphs are displayed in textual form, see Figure 7.2. First the quantities in the process structure graph are listed, followed by the relations. The list ends with the process frames and decomposition frames. A quantity is specified by the role, the quantity class and a unique name for the quantity. The role of a quantity can be input, output or internal. A relation is specified by the relation type and the quantities involved in the relation. A decomposition frame is specified by the predicate d-frame. A process frame is specified by the predicate p-frame. For each element a status code is shown that specifies how the element is used in the process structure graph. The possible status codes are described in Section 7.3.

Given a modelling question in terms of quantities of interest, exogenous quantities and irrelevant quantities, the Desimal system will construct all process structure graphs that represent a possible process decomposition for the phenomenon under study. The modeller has to accept one process structure graph as the appropriate process decomposition for the phenomenon under study. Different process structure graphs are constructed if the phenomenon under study is described in terms of one or more quantities that can be decomposed into subquantities. Selecting the appropriate process decomposition means that the modeller has to decide which quantities are describe as aggregate quantities and which quantities are described as compositions of other quantities.

After the modeller has selected the appropriate process structure graph, dy-



```
I_OUTPUT  : OUTPUT( [illegible]; TomatoColour )
EXOGENOUS : INPUT( [illegible]; Temperature )
RELEVANT  : INTERNAL( [illegible]( Chlorophyll_1 )
RELEVANT  : INTERNAL( [illegible]( Lycopene )
RELEVANT  : INTERNAL( ReactionRate( ReactionRate_1 )
RELEVANT  : INTERNAL( SimpleRate( SimpleRate_1 )
RELEVANT  : INTERNAL( ConsumedReactant( ConsumedReactant_1 )
RELEVANT  : INTERNAL( ApparentRate( ApparentRate_1 )
RELEVANT  : INTERNAL( ReactionRate( ReactionRate_2 )
RELEVANT  : TAD( Chlorophyll_1, TomatoColour )
RELEVANT  : TAD( Lycopene, TomatoColour )
RELEVANT  : AND( Chlorophyll_1, ConsumedReactant, DecayReaction_1 )
RELEVANT  : DCON( ReactionRate_1, Chlorophyll_1 )
RELEVANT  : AND( Lycopene, ProducedReactant, FormationReaction_1 )
RELEVANT  : DCON( SimpleRate_1, Lycopene )
RELEVANT  : DCON( ConsumedReactant_1, Lycopene )
RELEVANT  : AND( ReactionRate_1, SimpleRate, ReactionKinetics_1 )
RELEVANT  : FIAV( Temperature, ReactionRate_1 )
```

Figure 7.2: The first process structure graph for the phenomenon of colour development of tomatoes as function of temperature.

namic product models can be constructed that correspond with the frames in the selected process structure graph. This step may result in a number of dynamic product models, if the process structure graph contains one or more relations for which several quantitative models are applicable.

The dynamic product models are displayed as a list of submodel declarations. A submodel declaration specifies the name of the submodel, followed by a list of mappings between the model variables and the quantities in the process structure graph from which the dynamic product model is constructed. For example, the first submodel declaration in Figure 7.3 specifies that the model ExpDecrease is a submodel in the dynamic product model. The model variable k is mapped on the quantity ReactionRate_1. The model variable A is mapped on the quantity Chlorophyll. For each submodel a status code is shown that specifies how the submodel is used in the dynamic product model. The possible status codes are described in Section 7.3.

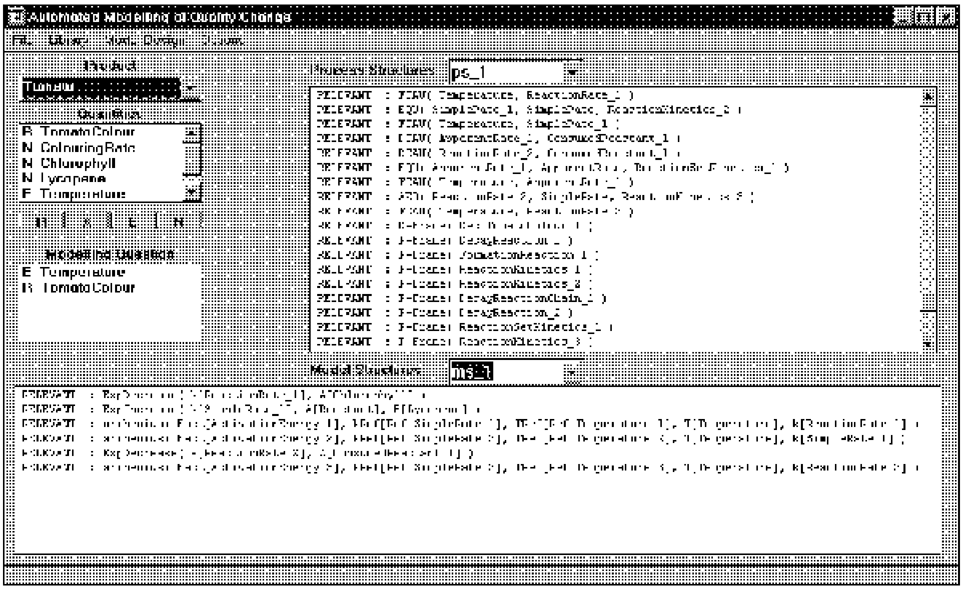
The modeller has to accept one dynamic product model as the appropriate quantitative model for the phenomenon under study. The DESIMAL system will generate a PROSIM model for the selected dynamic product model.

The generated PROSIM model for the dynamic product model displayed in Figure 7.3 consists of three modules and four macros. Each module and macro is written into separate files. These files have to be imported into the PROSIM environment to construct an executable PROSIM model. The main module is displayed in Figure 7.4. The main module contains a call to the macro initdata to initialise the constants in the model. This macro is not shown. After initialisation, the components are activated. The components Comp_1, Comp_2, Comp_3 and Comp_4 implement the processes and quantity decompositions. The component watch_mod generates an output file with the model results. This component is not shown. The other modules and the macros are displayed in Figure 7.5.

7.3 Architecture

The DESIMAL system has been implemented in Kappa-PC, a programming environment for knowledge based systems. The building of blocks of a knowledge bases system programmed in Kappa-PC are objects, methods, and rules.

Figure 7.6 displays the classes used in the implementation of the object model for the DESIMAL library. The modelling tasks are implemented as methods in



Product: Tomato

Qualities:
R TomatoColour
N ColouringRate
N Chlorophyll
N Lycopene
E Temperature

Modelling Question:
E Temperature
R TomatoColour

Process Structures: ps_1

```
RELEVANT  : FDAV( Temperature, ReactionRate_1 )
RELEVANT  : EQU( SimpleRate_1, SimpleRate, ReactionKinetics_2 )
RELEVANT  : FDAV( Temperature, SimpleRate_1 )
RELEVANT  : DDAV( ApparentRate_1, [illegible]_1 )
RELEVANT  : DDAV( ReactionRate_2, [illegible]_1 )
RELEVANT  : EQU( ApparentRate_1, ApparentRate, [illegible]_1 )
RELEVANT  : FDAV( Temperature, ApparentRate_1 )
RELEVANT  : EQU( ReactionRate_2, SingleRate, ReactionKinetics_3 )
RELEVANT  : FDAV( Temperature, ReactionRate_2 )
RELEVANT  : P-Frame( [illegible] 1 )
RELEVANT  : P-Frame( DecayReaction_1 )
RELEVANT  : P-Frame( FormationReaction 1 )
RELEVANT  : P-Frame( ReactionKinetics 1 )
RELEVANT  : P-Frame( ReactionKinetics_2 )
RELEVANT  : P-Frame( DecayReactionChain_1 )
RELEVANT  : P-Frame( DecayReaction_2 )
RELEVANT  : P-Frame( ReactionSetKinetics_1 )
RELEVANT  : P-Frame( ReactionKinetics_3 )
```

Model Structures: ms_1

```
[illegible]  : ExpFunction( [illegible]_1], A[Chlorophyll] )
[illegible]  : ExpFunction( [illegible]_1], A[Product], B[Lycopene] )
[illegible]  : Arrhenius Eq( [ActivationEnergy 1], Rref[Ref SingleRate 1], Tref[Ref Temperature 1], T[Temperature], k[ReactionRate 1] )
[illegible]  : Arrhenius Eq( [ActivationEnergy 2], Rref[Ref SingleRate 2], Tref[Ref Temperature 2], T[Temperature], k[SimpleRate 1] )
[illegible]  : ExpDecrease( [ReactionRate 2], [illegible] 1] )
[illegible]  : Arrhenius Eq( [ActivationEnergy 3], Rref[Ref SingleRate 3], Tref[Ref Temperature 3], T[Temperature], k[ReactionRate 2] )
```

Figure 7.3: The model structure created for the process structure graph.

```
@ ===================
@ Module : MOD MAINMOD
@ ===================
@ CALL initdata

ACTIVATE comp_1 FROM Start_comp_1
ACTIVATE comp_2 FROM Start_comp_2
ACTIVATE comp_3 FROM Start_comp_3
ACTIVATE comp_4 FROM Start_comp_4
ACTIVATE WATCHER FROM start_watch IN WATCH_MOD

WAIT MAXTIM

CANCEL ALL
TERMINATE

Start_comp_1:
SPECIFY qTomatoColour
    PRECEPT(qTomatoColour := Sum( qChlorophyll, qLycopene ))

Start_comp_2:
SPECIFY qChlorophyll
    PRECEPT(qChlorophyll := ExpDecrease( qReactionRate_1, qChlorophyll ))
SPECIFY qReactionRate_1
    PRECEPT(qReactionRate_1 := arrhenius( qActivationEnergy_1, qRef_SimpleRate_1,
                                          qRef_Temperature_1,qTemperature ))

Start_comp_3:
SPECIFY qLycopene
    PRECEPT(qLycopene := ExpIncrease( qSimpleRate_1, qReactant ))
SPECIFY qSimpleRate_1
    PRECEPT(qSimpleRate_1 := arrhenius( qActivationEnergy_2, qRef_SimpleRate_2,
                                        qRef_Temperature_2, qTemperature ))

Start_comp_4:
SPECIFY qConsumedReactant_1
    PRECEPT(qConsumedReactant_1 := ExpDecrease( qReactionRate_2, qConsumedReactant_1 ))
SPECIFY qReactionRate_2
    PRECEPT(qReactionRate_2 := arrhenius( qActivationEnergy_3, qRef_SimpleRate_3,
                                          qRef_Temperature_3, qTemperature ))
```

Figure 7.4: The main module of the Prosim module contains the calls to the submodels.

```
@ ===================
@ Module : MOD DEFINE
@ ===================
COMPONENT:
 WATCHER
 Comp_1
 Comp_2
 Comp_3
 Comp_4

ATTRIBUTE OF Comp_1 :
 CONTINUOUS(1) : qTomatoColour
 CONTINUOUS(1) : qChlorophyll
 CONTINUOUS(1) : qLycopene

ATTRIBUTE OF Comp_2 :
 CONTINUOUS(1) : qReactionRate_1
 REAL          : qActivationEnergy_1
 REAL          : qRef_SimpleRate_1
 REAL          : qRef_Temperature_1

ATTRIBUTE OF Comp_3 :
 CONTINUOUS(1) : qSimpleRate_1
 CONTINUOUS(1) : qConsumedReactant_1
 CONTINUOUS(1) : qReactant
 REAL          : qActivationEnergy_2
 REAL          : qRef_SimpleRate_2
 REAL          : qRef_Temperature_2

ATTRIBUTE OF Comp_4 :
 CONTINUOUS(1) : qApparentRate_1
 CONTINUOUS(1) : qReactionRate_2
 REAL          : qActivationEnergy_3
 REAL          : qRef_SimpleRate_3
 REAL          : qRef_Temperature_3

ATTRIBUTE OF WATCHER :
 REAL          : WATCH_FREQ

ATTRIBUTE OF MAIN :
 REAL          : MAXTIM
 REAL          : qTemperature

INPUTSTREAM    : DATIN
OUTPUTSTREAM : DATOUT
```

```
@ ===================
@ Macro : MAC ExpDecrease
@ ===================
PARAMETERS :
   REAL: k
   REAL: A

RETURN 0 - k*A

@ ===================
@ Macro : MAC arrhenius
@ ===================
PARAMETERS :
   REAL: Eact
   REAL: kRef
   REAL: TRef
   REAL: T

LOCALS :
   REAL: Tabs

Tabs := T + 273
RETURN kRef * exp( Eact * ((1/TRef) - (1/Tabs)))

@ ===================
@ Macro : MAC ExpIncrease
@ ===================
PARAMETERS :
   REAL: k
   REAL: A

RETURN k*A
```

Figure 7.5: The mod define module and the macros in the generated PROSIM model.

these classes.

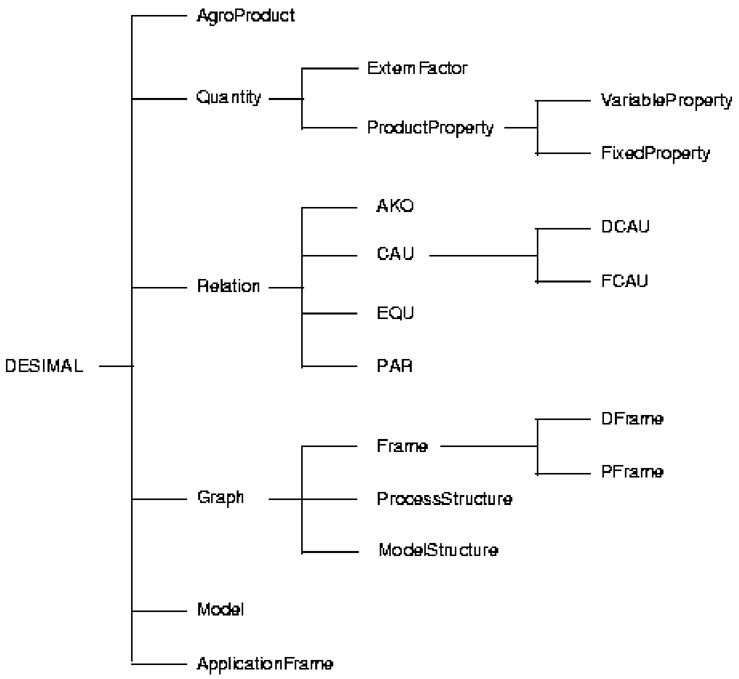


Figure 7.6: The class hierarchy of the DESIMAL system.

- The class **AgroProduct** is the root for a hierarchy of agricultural products. For each agricultural product, an instance of this class is created that holds the quality attributes and the product properties that are relevant for the agricultural product. This knowledge is used to restrict the search algorithm used in Qualitative Process Analysis.

- The class **Quantity** is the base class for the external factors, the fixed product properties and the variable product properties. Each **Quantity** instance has a dimension and a role. The dimension (unit of measurement) can be used to automatically generate interactions between variables of mathematical models. The role of a quantity specifies whether the quantity is an input quantity, an internal quantity or an output quantity of a process structure graph, and is assigned during Qualitative Process Analysis.

- The class Relation is the root for the knowledge graph relations between quantities in the knowledge graph frames. The AKO, EQU, DCAU, FCAU, and PAR relations are explicitly represented as subclasses of the Relation class. Each Relation instance connects two Quantity instances, called the source and the target of the relation. As the AKO and EQU relations connect quantities in different frames, the instances of the corresponding classes additionally specify the Frame instance the target quantity belongs to.
- The class Graph is the base class for the directed graphs that are used in the DESIMAL system. Each instance of this class has a list of quantities and relations that make up the graph.
- The class ProcessStructure is a subclass of Graph representing the process structure graphs that are constructed during Qualitative Process Analysis. Besides the quantities and relations, an instance of this class also contains a list of Frame instances that are included in the process structure graph.

  Furthermore, each ProcessStructure instance maintains information about the current status of the quantities, relations and frames in the Qualitative Process Analysis task. The following status codes are used for quantities, relations and frames in a ProcessStructure instance:

  - exogenous

    Only quantities can have an exogenous status. This status specifies that the quantity is an input of the process structure graphs. Effects on quantities with an exogenous status are not represented in the process structure graph.
  - q_output

    Only quantities can have a q_output status. This status specifies that the quantity is a quantity of interest. The process structure graph must describe the effects on such quantities.
  - irrelevant

    A quantity with an irrelevant status is specified as, or found to be, irrelevant. Neither the effects of this quantity on quantities of interest, nor the effects on this quantity are represented in the process structure graph.

    The irrelevant status is also assigned to PAR relations in a decomposition frame for an aggregate quantity in a process structure graph that describes the aggregate quantity at the aggregate level. The decomposition frame has to be included in this process structure graph, because the frame may contain AKO and EQU relations from the aggregate quantity to quantities in other frames.

  - possible

    Quantities, relations and frames may get a possible status during the first step of Qualitative Process Analysis: retrieval of relevant frames.

    A possible status for a quantity indicates that the quantity affects one of the quantities of interest. A relation gets a possible status when it is found to represent an effect on a quantity of interest. A frame gets a possible status when one of the relations in the frame is found to represent an effect on a quantity of interest.

  - relevant

    Quantities, relations and frames may get a relevant status during the second step of Qualitative Process Analysis: selection of adequate process structure graphs.

    A relevant status for a quantity indicates that the quantity is on a relevant influence path from an exogenous quantity to a quantity of interest. The relations that make up a relevant influence path also have a relevant status, as well as the frames the relations belong to.

- The class ModelStructure is a subclass of Graph representing the model structure graphs that are constructed during Simulation Model Construction. The nodes in a model structure graph are Model instances. The arcs in a model structure graph are Interaction instances.

- The class Frame is the root for the knowledge graph frames that make up the qualitative level of the DESIMAL library. Each frame is a small directed graph. The nodes are Quantity instances. The arcs are Relation instances. The DCAU, FCAU, and PAR relations are connections between quantities in one frame. The AKO and EQU relations connect two quantities in two frames.

- The class ApplicationFrame represents the application frames that specify the relationships between the mathematical models and the knowledge graph frames. An application frame specifies the knowledge graph frames for which a mathematical model is applicable. It also specifies a mapping between the variables in the mathematical model and quantities in these knowledge graph frames.

- Model

  The class Model is the root class of the library of mathematical models. Each Model instance represents a mathematical model in the DESIMAL library. A Model subclass has a list of equations, and lists of input, local and output variables. Each equation must be an assignment of input and internal variables to one output variable of the model.

  The constructed simulation model may have several ApplicationFrame instances for one Model instance, representing multiple applications of one model in the simulation model. These ApplicationFrame instances represent mappings of the model variables to different quantities in the process structure graph.

The explicit representation of the knowledge graph relations as objects is most useful for AKO relations. These relations represent generalisation relationships between quantities, and thus could be implemented by the inheritance capabilities of Kappa-PC. However, as one quantity may have AKO relations to several other quantities, this would require multiple inheritance, which is not supported by Kappa-PC.

More importantly, the AKO relations not only represent generalisation relationships, but also represent modelling assumptions. Explicitly representing the AKO relations enables to reason about the modelling assumptions. This would not be possible when the AKO relations were implemented using the inheritance capabilities of an object hierarchy. Moreover, the modelling assumptions need not correspond with inheritance relationships between quantities.

An additional advantage of the explicit representation of the knowledge graph relations as objects, is that the object model closely corresponds with the knowledge graph formalism presented in Chapter 5.

7.4 Generation of the PROSIM models

PROSIM is a language to develop process-oriented models both for discrete-time and continuous-time simulation. The DESIMAL system uses only the continuous-time capabilities.

The building blocks of a PROSIM model are called components. The components are executed in parallel and can be activated and deactivated dynamically. Therefore, this language is well-suited to implement and execute the

Dynamic Product Model specifications constructed in the Simulation Model Construction task.

A component in a continuous-time PROSIM model consists of a set of attributes, a set of algebraic and differential equations for these attributes, and an INTEGRATE statement. The attributes can have the following types:

- REAL
  Attributes of this type have a constant value throughout a simulation run. This type is used for fixed properties.

- CONTINUOUS(0)
  Attributes of this type change instantaneously. This type is used for variable properties that are influenced by FCAU relations.

- CONTINUOUS(1)
  Attributes of this type do not change instantaneously. This type is used for variable properties that are influenced by DCAU relations.

Higher orders are possible for CONTINUOUS attributes, but are not used in DESIMAL.

A PROSIM implementation of a Dynamic Product Model consists of the components MAIN and WATCHER, and has one or more *process components* implementing the quantitative relations. The component MAIN is required in any PROSIM model, and is active throughout the simulation run. This component is typically used to activate other components and to control when the simulation model has to terminate. The component WATCHER at regular intervals writes the values of the model variables to an output file.

The process components implement the quantitative relations of the Dynamic Product Model. A process component may contain one or more quantitative models, but cannot contain a part of a quantitative model. The boundaries between process components are formed by differences in occurrence conditions of the processes described in the Process Structure Graph. Each process component implements a piece of the phenomenon's behaviour that can be activated and deactivated. Therefore, interacting processes that have equal occurrence conditions are implemented in one process component in the PROSIM model. Consequently, if all processes have the same occurrence conditions, then the generated PROSIM model will contain only one process component.

The boundaries of the process components are determined by examining the qualitative relations on the input quantities of the knowledge graph frames. If an input quantity of a process frame or decomposition frame is the target of a DCAU relation, then the quantity is influenced by another process. If the knowledge graph frame for this influencing process has an occurrence condition that differs from the process that is influenced by the quantity, then these processes are implemented in separate process components, so that the processes can be activated independently.

The specification of a PROSIM model consists of a number of modules and macros. The modules MOD DEFINE and MOD MAINMOD are required in each PROSIM model specification. Apart from these modules, a PROSIM model specification generated by DESIMAL contains a module MOD WATCH_MOD, a macro MAC INITDATA, and a number of modules and macros to implement the quantitative relations of the model.

- MOD DEFINE
  This module is required in any PROSIM model. It contains declarations of the components in the PROSIM model and of the attributes of these components.

- MOD MAINMOD
  This module is required in any PROSIM model. It contains the specification of the component MAIN. It may also contain the specification of other components.

- MOD WATCH_MOD
  This module contains the specification of the component WATCHER.

- MAC INITDATA
  This macro is called from MOD MAINMOD to initialise the simulation model to read the values for the fixed properties, the initial values for the variable properties, and the scenarios for the external factors.

The specification of a process component consists of a set of SPECIFY statements and at least one INTEGRATE statement. The SPECIFY statements are used both for algebraic equations and for differential equations. The equations are executed with the INTEGRATE statement, even if the component contains only algebraic equations. In the PROSIM models generated by the DESIMAL system, the SPECIFY statements have the following form:

```
SPECIFY $v1$ PRECEPT $(v1 \leftarrow m1(p1, \ldots, pm))$
SPECIFY $v2$ PRECEPT $(v2' \leftarrow m2(q1, \ldots, qn))$
```

where *v1* is a CONTINUOUS(0) variable, *m1* is a macro implementing the algebraic equation for *v1*, *v2* is a CONTINUOUS(1) variable, and *m2* is a macro implementing the differential equation for *v2*.

A process component may describe a process for which several models are applicable. As was described in Section 6.5.2, in that case a number of consistent simulation models are generated. A consistent simulation model contains exactly one equation for each knowledge graph relation. In general, the specification of a process component contains of a group of SPECIFY statements and an INTEGRATE statement for each consistent simulation model. This consistent simulation model is used as long as the operating conditions of all the models in the consistent simulation model are satisfied. If the operating condition of one of the models is not satisfied any more, then another consistent simulation model is selected. The component terminates if no consistent simulation model can be selected. In general, the specification of a process component has the following structure:

```
start_component:

IF opcond(ma1) AND opcond(mb)
    SPECIFY $A$ PRECEPT($A' \leftarrow ma1$)
    SPECIFY $B$ PRECEPT($B' \leftarrow mb$)
    INTEGRATE WHILE (opcond(ma1) AND opcond(mb))
    REPEAT FROM start_component
END

IF opcond(ma2) AND opcond(mb)
    SPECIFY $A$ PRECEPT($A' \leftarrow ma2$)
    SPECIFY $B$ PRECEPT($B' \leftarrow mb$)
    INTEGRATE WHILE (opcond(ma2) AND opcond(mb))
    REPEAT FROM start_component
END
```

This process component is active as long as the operating conditions for the model mb, and the operating conditions of either the model ma1 or the model ma2 are satisfied. The process component will terminate if the simulation state does not satisfy the operating conditions of the models that are selected to describe the process. In that case, the simulation model is used outside its foreseen operating region, and has to be stopped.

7.5 The process library

The idea behind the DESIMAL method is that the physiological phenomena underlying quality change can be decomposed into a set of generic processes. In this section the library of generic processes as it is used in the DESIMAL system is described.

The DESIMAL library provides a collection of generic frames each representing a specific behaviour often encountered in physiological phenomena underlying quality change of agricultural products.

7.5.1 Chemical reactions

Chemical reactions are ubiquitous in modelling physiological phenomena. The DESIMAL library contains frames for several types of chemical reactions, and contains a collection of generic quantities involved in these frames. The quantities Reactant, ConsumedReactant and ProducedReactant represent the concentrations of substances involved in the chemical reaction. A chemical reaction is decomposed into a formation reaction of the produced reactant and a decay reaction of the consumed reactant.

Each chemical reaction in some way depends on the temperature, according to Arrhenius' law or a function of Arrhenius' law. The quantities SimpleRate and ApparentRate represent the rate of a chemical process. ReactionKinetics and ReactionSetKinetics are generic frames to represent the dependency of a chemical process on the temperature.

7.5.1.1 Quantities

Quantity	Description
Reactant	An abstract quantity representing a substance that is either consumed or produced in a chemical reaction or is modelled as such.
ConsumedReactant	A specialisation of Reactant, representing a substance that is (modelled as) consumed by a chemical reaction.
ProducedReactant	A specialisation of Reactant, representing a substance that is (modelled as) produced by a chemical reaction.
SimpleRate	Represents the rate of exactly one chemical reaction.
ApparentRate	Represents the apparent rate of a set of chemical reactions, and is an explicit or implicit composition of at least two specialisations of SimpleRate or ApparentRate.

7.5.1.2 Frames

- FormationReaction
  This is a generic process that represents that the increase in the quantity ProducedReactant is caused by one chemical reaction. The rate of the reaction is represented by the quantity SimpleRate. The source of the reaction is represented by the quantity Reactant, representing that the source is assumed to be abundant with respect to the reaction, and will, therefore, not be affected by the reaction. However, if the reaction has an effect on the source, then the source has to be represented by the specialised quantity ConsumedReactant. The decrease has to be described by the process frame DecayReaction with the same SimpleRate.

  Relations: dcau(SimpleRate, ProducedReactant)
  dcau(Reactant, ProducedReactant)

- DecayReaction
  This generic process represents that the decrease in the quantity ConsumedReactant is caused by one chemical reaction. The rate of the reaction is represented by the quantity SimpleRate.

Relations: dcau(SimpleRate, ConsumedReactant)

- FormationReactionChain
  This generic process has to be used when the increase in the quantity ProducedReactant is the result of a (possibly assumed) chain of chemical reactions. The intermediate reaction rates are combined into a specialisation of the generic quantity ApparentRate. The source of the reaction chain is represented by the quantity Reactant, representing that the source is assumed to be abundant with respect to the reaction chain, and will, therefore, not be affected by the reaction chain. However, if the reaction chain has an effect on the source, then the source has to be represented by the specialised quantity ConsumedReactant. The decrease has to be described by the process frame DecayReactionChain with the same ApparentRate.

  Relations: dcau(ApparentRate, ProducedReactant)
  dcau(Reactant, ProducedReactant)

- DecayReactionChain
  This generic process models the decrease in the quantity ConsumedReactant as the result of a (possibly assumed) chain of chemical reactions. The intermediate reaction rates are combined into a specialisation of the generic quantity ApparentRate.

  Relations: dcau(ApparentRate, ConsumedReactant)

- ReactionKinetics
  This generic process frame relates the quantity SimpleRate to the external factor Temperature. Generally, Arrhenius' law is used as a quantitative model for this process frame.

  Relations: fcau(Temperature, SimpleRate)

- ReactionSetKinetics
  This generic process frame relates the quantity ApparentRate to the external factor Temperature. The temperature dependency is generally a function of a number of Arrhenius' laws, one for each reaction rate that is assumed to be a part of the ApparentRate.

  Relations: fcau(Temperature, ApparentRate)

7.5.2 Catalysis reactions

An important type of chemical reactions are the catalysis reactions. In such reactions, one reactant serves as a trigger for the reaction. In normal catalysis reactions, the catalyst is not influenced by the reaction. However, autocatalysis reactions cause a net increase of the catalyst.

7.5.2.1 Quantities

Quantity	**Description**
Catalyst	The quantity that triggers the catalysis reaction. The quantity Catalyst is a specialisation of the generic quantity Reactant.
AutoCatalyst	The quantity that triggers the autocatalysis reaction and is increased by the reaction. The quantity AutoCatalyst is a specialisation of the generic quantity Catalyst.

7.5.2.2 Frames

- Catalysis
  Catalysis is a chemical reaction that only occurs when the quantity Catalyst has a non-zero concentration. Enzymatic reactions are an important class of catalysis processes. The Catalyst is both consumed and produced by the process, but the net concentration of the Catalyst is not changed by the process. Therefore, the frame does not have relations for the behaviour of the quantity Catalyst. As the rate of a catalyst reaction in general cannot be described by a simple Arrhenius function, the rate is represented by the quantity ApparentRate. The source of the catalysis reaction is represented by the quantity Reactant, representing that the source is assumed to be abundant with respect to the reaction, and will, therefore, not be affected by the reaction. However, if the catalysis reaction has an effect on the source, then the source has to be represented by the specialised quantity ConsumedReactant. The decrease has to be described by a process frame that relates the quantity to the ApparentRate in the process frame Catalysis.

Relations: dcau(ApparentRate, ProducedReactant)
dcau(Catalyst, ProducedReactant)
dcau(Reactant, ProducedReactant)

- AutoCatalysis
  A generic process representing that the increase in the quantity AutoCatalyst is the result of an autocatalytic reaction. This process frame is a qualitative representation of a logistic function, with a rate represented by the quantity AC_Rate. The quantities AC_Minimum and AC_Maximum represent the range of the quantity AutoCatalyst.

  Relations: dcau(AC_Rate, AutoCatalyst)
  dcau(AC_Minimum, AutoCatalyst)
  dcau(AC_Maximum, AutoCatalyst)

7.5.3 Empirical relations

The above process of AutoCatalysis is a qualitative model for the logistic function. This is an empirical function: the behaviour of the quantity follows a pattern of behaviour, but the pattern of behaviour cannot be explained in terms of underlying processes. In such cases an empirical function, such as the logistic function, is used. An other example of such an empirical function is limited exponential growth, represented by the quantities and the frame below.

7.5.3.1 Quantities

Quantity	Description
LE_Quantity	The quantity that changes according to a limited exponential growth function.
LE_Minimum	The lower limit for LE_Quantity.
LE_Maximum	The upper limit for LE_Quantity.

7.5.3.2 Frames

- LimitedExponential
  A generic process representing that the quantity LE_Quantity changes exponentially in a range represented by the quantities LE_Minimum and

LE_Maximum. These quantities have FCAU relations with the quantity LE_Quantity because the latter quantity will change immediately when the lower or upper limits change. The rate is represented by the quantity LE_Rate.

Relations: dcau(LE_Rate, LE_Quantity)
fcau(LE_Minimum, LE_Quantity)
fcau(LE_Maximum, LE_Quantity)

7.6 Cases

In the rest of this chapter a number of modelling cases are elaborated for which simulation models have been developed by the DESIMAL system.

7.6.1 Rate of chemical reactions

Many physiological processes in perishable products are systems of chemical reactions. The rate of each chemical reaction in the system depends on temperature, according to Arrhenius' law [Chang, 1981]. The complexity of the reaction system underlying a physiological process can partly be derived from the process rate. If the process rate is monotonically increasing with temperature, one chemical reaction may be assumed to underlie that process.

If the rate shows a maximum or minimum at some temperature, several reactions have to be assumed to underlie the process. Such a combination of chemical reaction rates in principle should be modelled as a combination of Arrhenius functions. However, one Arrhenius function may be used if the process rate decreases or increases monotonically with increasing temperature.

7.6.1.1 Types of knowledge

This case is about one *behavioural relation*: the dependency of the rate of a chemical process on the temperature. This behavioural relation has different mathematical formulations depending on the *modelling assumption* made for the process rate: as a rate of a simple chemical reaction or as a combination of such reaction rates.

The case includes *decomposition knowledge* about the apparent rate of a complex chemical process: the apparent rate can either be modelled as a black box quantity or as the composition of the reaction rates of the individual chemical reactions in the process.

7.6.1.2 Knowledge graph

To describe the effect of the temperature on the rate of a chemical process the modeller has to determine whether the rate of the process can be modelled as a simple chemical rate, or as a composition of several rates. Depending on which modelling assumption is made, one of the two process frames displayed in Figure 7.7 is used.

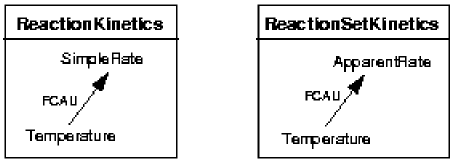


Figure 7.7: Knowledge graphs for the rate of a chemical process.

The process frame ReactionKinetics represents the relation between the temperature and the rate of one chemical reaction, which is represented by the quantity SimpleRate. If a process rate is assumed to be a simple chemical rate, the process rate has to be a specialisation of SimpleRate, and an instance of the process frame ReactionKinetics has to be created to model the effect of the temperature on the process rate.

The process frame ReactionSetKinetics represents the relation between the temperature and the apparent rate of a complex of chemical reactions. The quantity ApparentRate represents the composition of the rates of these chemical reactions. The DESIMAL library does not contain a decomposition frame for ApparentRate, because the decomposition of an apparent rate depends on the phenomenon under study, and therefore has to be represented in a phenomenon specific decomposition frame. For example, the Michaelis-Menten model for the kinetics of enzymatic processes contains a specificity constant (K_m) that is a composition of the reaction rates of the three reactions involved in the enzymatic process [Chang, 1981].

7.6.2 Colouring of tomatoes

The colour of a tomato is a quality attribute that is the user's perception of the amounts of two colouring substances, chlorophyll and lycopene. Chlorophyll is responsible for the green colour of an unripe tomato, and lycopene gives rise to the red colour of a ripe tomato. During ripening under normal temperatures (between 12°C and 25°C) chlorophyll degrades, and lycopene is formed. The degradation of chlorophyll and the forming of lycopene are assumed to be two enzymatic processes [Grierson and Kader, 1986; Shewfelt *et al.*, 1988]. The enzyme involved in the chlorophyll degradation is assumed to be inactive at low temperatures, leading to discolouring when the tomatoes are stored at too low temperatures. Discolouring at high temperatures is assumed to be caused by a denaturation occurring at high temperatures of the enzyme involved in the lycopene forming process. In [Tijskens and Evelo, 1994] the combination of chlorophyll degradation and lycopene forming is modelled as one logistic function, with a temperature-dependent rate that is a combination of the chlorophyll degradation rate and the lycopene forming rate.

7.6.2.1 Knowledge graph

This case contains a *quality assignment relation* between the tomato colour (quality attribute) and the chlorophyll and lycopene concentrations (product properties). This quality assignment relation is represented by the decomposition frame Dec-TomatoColour in Figure 7.8. This decomposition frame is the link between the two levels of detail at which the tomato colour is described.

At the aggregate level, the tomato colour is assumed to be the result of a chain of reactions. This *modelling assumption* is specified by the AKO relations between the quantities in the process frames TomatoColouring and FormationReactionChain in the upper part of the knowledge graph displayed in Figure 7.8.

At the detailed level, the changes in the tomato colour are described as resulting from changes in the amounts of the individual colouring substances, Chlorophyll and Lycopene. The AKO relation from Chlorophyll to ConsumedReactant in DecayReaction represents the modelling assumption that the chlorophyll conversion can be modelled as a chemical degradation. The AKO relation from Lycopene to ProducedReactant in FormationReaction represents another modelling assumption that the lycopene production can be modelled as a chemical formation.

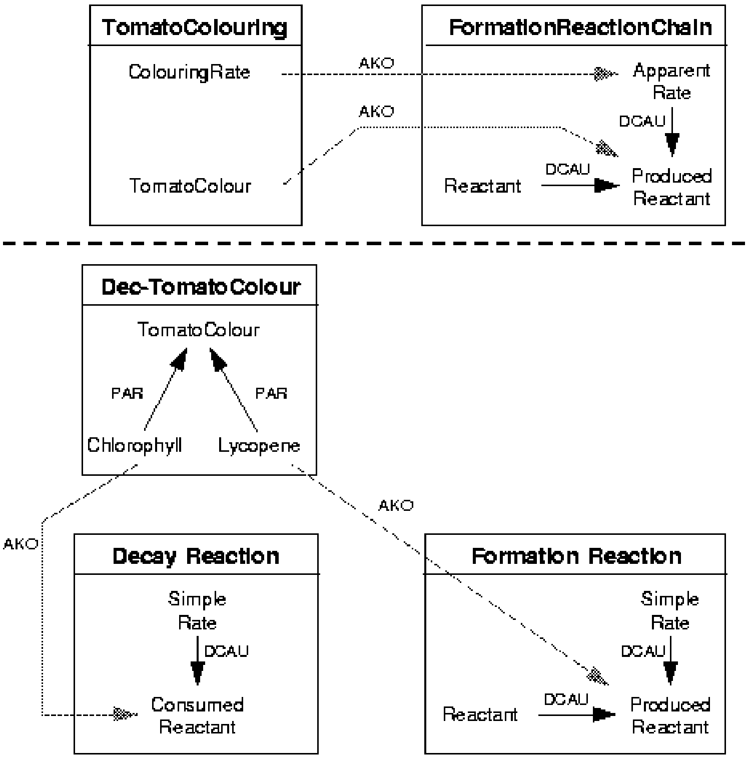


Figure 7.8: The knowledge graph about colouring of tomatoes contains two levels of detail at which this phenomenon can be modelled. These levels are separated by the dashed line.

7.6.2.2 Construction of the process structure graphs

To illustrate the construction of the process structure graphs for the colouring of tomatoes, suppose that the modelling question specifies TomatoColour as the quantity of interest, and Temperature as the exogenous quantity. Furthermore, suppose that the tomatoes are not packaged, so that influences of the tomatoes on the temperature can be neglected.

To find process structure graphs for this modelling question, Qualitative Process Analysis starts with an initial process structure graph, containing only the quantities TomatoColour and Temperature.

In the first step of the retrieval task, the decomposition frame Dec-

TomatoColour is found to be possibly relevant. Hence, the initial process structure graph is duplicated. In the one graph, the decomposition is ignored, so that the quantity TomatoColour is described at the aggregate level. In the other graph, the decomposition is taken into account, describing the TomatoColour as the composite of the colouring substances Chlorophyll and Lycopene.

After the first step, the Qualitative Process Analysis continues with the first, aggregate level, process structure graph. The subsequent steps for the refinement of the first process structure graph are displayed in Figure 7.9. Part (A) in this figure displays the quantities in the modelling question.

- In the second step, the process frame TomatoColouring is found to have an effect on the quantity TomatoColour, and is therefore added to the process structure graph. (see Figure 7.9 (B)).
- The third step considers the AKO relations in the process frame TomatoColouring. An instance of the process frame FormationReactionChain is created to describe the behaviour of TomatoColour as the result of a chain of chemical reactions (Figure 7.9 (C)).
- The fourth step considers the EQU relation for ApparentRate between the process frames FormationReactionChain and ReactionSetKinetics. An instance of the latter process frame is created to relate the rate of tomato colouring to temperature. This step results in the final process structure graph displayed in Figure 7.9 (D).

After completion of the first graph, Qualitative Process Analysis continues with the second, detailed level, process structure graph. The subsequent steps for the refinement of this process structure graph are displayed in Figure 7.10. Again, part (A) in this figure displays the quantities in the modelling question.

- In the first step, the decomposition frame Dec-TomatoColour is added to the graph (Figure 7.10 (B)). The second step, finding process frames for the quantity TomatoColour, is skipped, because the behaviour of this quantity has to be described at a more detailed level as a composition of the colouring substances.
- The third step considers the AKO relations in the decomposition frame. An instance of the process frame DecayReaction is created that models Chlorophyll as a kind of ConsumedReactant. In the same manner,

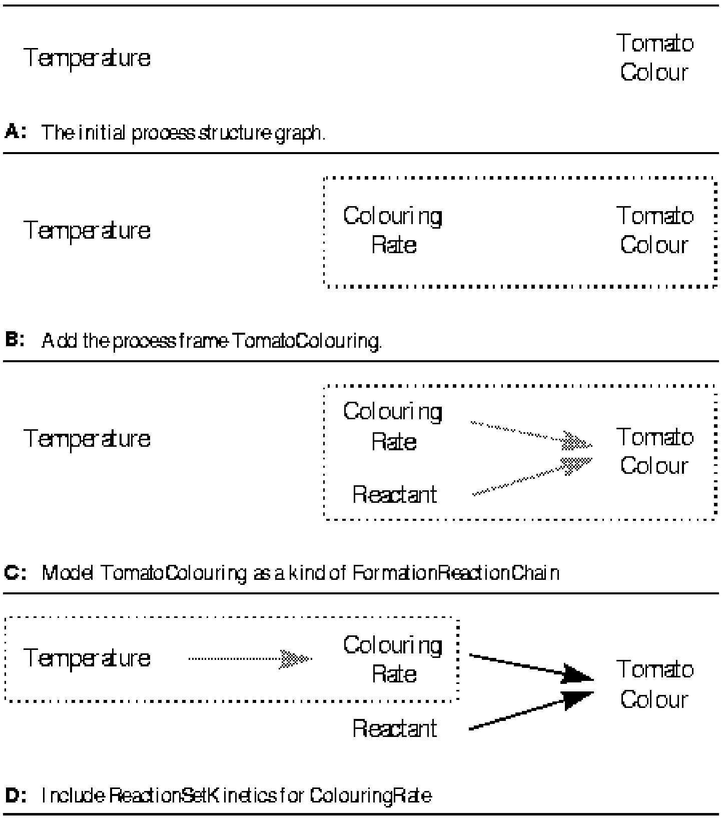
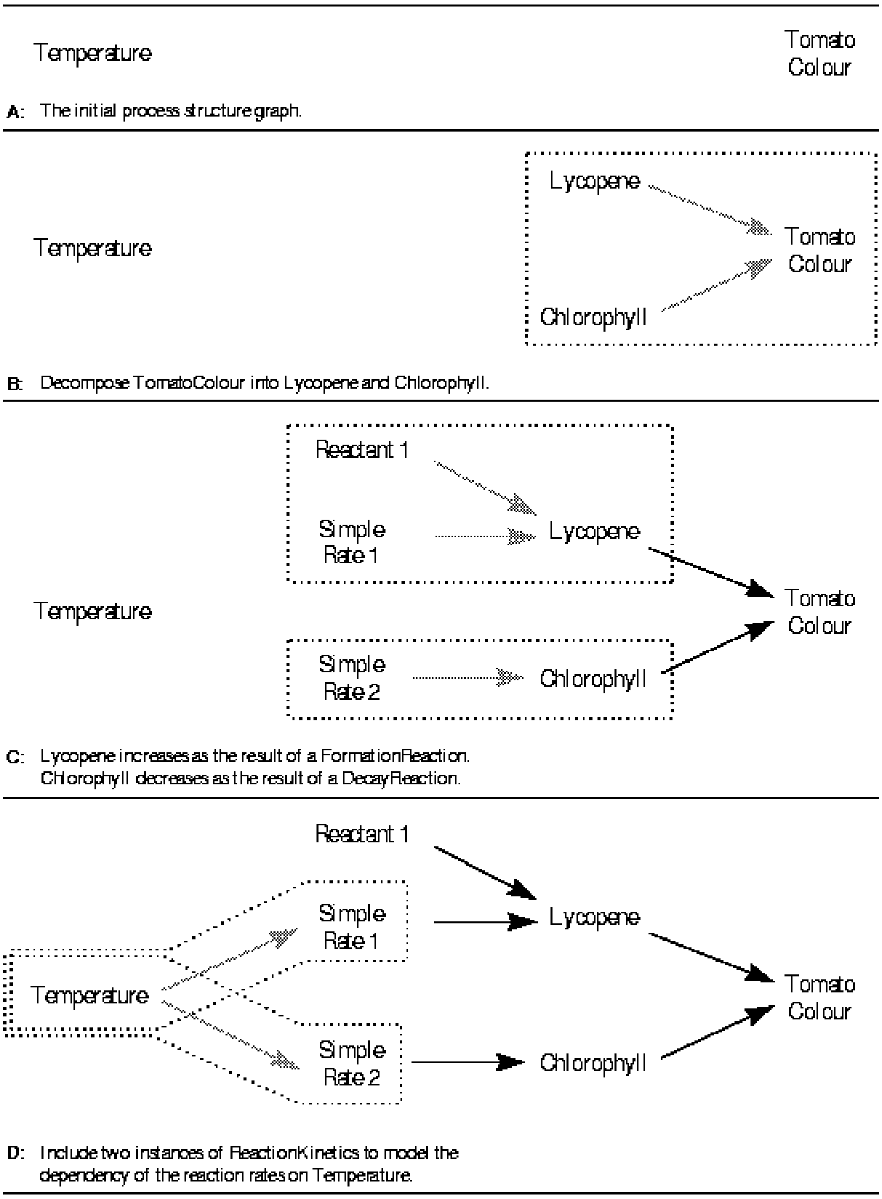

Temperature
Tomato Colour
A: The initial process structure graph.
Temperature
Colouring Rate
Tomato Colour
B: Add the process frame TomatoColouring.
Temperature
Colouring Rate
Reactant
Tomato Colour
C: Model TomatoColouring as a kind of FormationReactionChain
Temperature
Colouring Rate
Reactant
Tomato Colour
D: Include ReactionSetKinetics for ColouringRate

Figure 7.9: Construction of a process structure graph in which the colour of tomatoes is modelled as an aggregate quantity. Each part displays an extension of the process structure graph with the quantities and relations that appear in bold face. Boxes represent frames. See text for details.



A: The initial process structure graph.

B: Decompose TomatoColour into Lycopene and Chlorophyll.

C: Lycopene increases as the result of a FormationReaction. Chlorophyll decreases as the result of a DecayReaction.

D: Include two instances of ReactionKinetics to model the dependency of the reaction rates on Temperature.

Figure 7.10: Construction of a process structure graph in which the colour of tomatoes is modelled as a composition of the constituent colouring substances. Each part displays an extension of the process structure graph with the quantities and relations that appear in bold face. Boxes represent frames. See text for details.

an instance of the process frame FormationReaction is created in which Lycopene is modelled as a kind of ProducedReactant. Figure 7.10 (C) displays the resulting graph.

Note that the quantity ConsumedReactant in the latter process frame is not replaced by Chlorophyll, as these quantities are not connected by AKO relations in the knowledge graph displayed in Figure 7.8. The conversion of chlorophyll and the forming of lycopene are separate reactions.

- The fourth step considers the EQU relations for the quantities added to the process structure graph constructed so far. Again, the EQU relation for the quantity TomatoColour from the decomposition frame to the process frame TomatoColouring is ignored, because the behaviour of TomatoColour is described at a more detailed level as a composition of the colouring substances.

  The quantity SimpleRate in DecayReaction has an EQU relation with the same quantity in the process frame ReactionKinetics. Therefore, an instance of the latter process frame is created to represent that the rate of the degradation reaction depends on temperature. In the same manner, a separate instance of the process frame ReactionKinetics is created for the SimpleRate in FormationReaction.

  This step results in the final process structure graph displayed in Figure 7.10 (D).

7.6.3 Chilling injury

Chilling injury is a general term for visible forms of damage that may occur when products are stored at too low temperatures. The injury normally appears after a chilling period, when the product may already be stored at optimal conditions. This deferred appearance makes chilling injury difficult to comprehend and to model. By decomposing the phenomenon of chilling injury into generic processes, a model has been developed for the occurrence of chilling injury in cucumber fruits and bell peppers [Tijskens *et al.*, 1994]. The decomposition was based on the following assumptions:

- Chilling injury is the visible effect of too many free radicals that are generated by reactions in the living cells of the product. The occurrence of chilling injury was modelled as a chemical reaction that consumes the generated radicals. The generation of radicals was modelled as an autocatalytic process with respect to the amount of free radicals.

- At normal conditions no chilling injury is observed, so that the free radicals must be removed or inactivated in some way. This radical scavenging process was assumed to be an enzymatic process, affected by the amount of free radicals and by the enzyme activity.

- The enzyme in the radical scavenging process was assumed to denaturate irreversibly at low temperatures. This accounts for the fact that chilling injury only occurs after a period of too low temperatures.

By making the above assumptions and by using generic processes, it proved possible to develop a quantitative simulation model for the complex phenomenon of chilling injury. This model also correctly explained chilling injury phenomena that were not accounted for in the development of the simulation model, which proves the validity of the approach.

7.6.3.1 Knowledge graph

Figure 7.11 displays the knowledge graph for the chilling injury phenomenon. The product properties that are involved in this phenomenon are grouped in the process frame ChillingProcess. The relations between these properties are specified in the frames to which the product properties have AKO relations. The phenomenon of chilling injury is a complex interaction of four processes:

- The AKO relation from Radicals to AutoCatalyst in the process frame AutoCatalysis represents the modelling assumption that the increase of free radicals is an autocatalytic process.

- The AKO relation from Radicals to ConsumedReactant in the process frame ChemicalReaction represents the modelling assumption that the visible effect of free radicals on cell membranes (i.e. chilling injury) can be modelled as a conversion of free radicals to produce chilling injury.

- The AKO relation from Radicals to Substrate in EnzymaticReaction represents the modelling assumption that the scavenging of the free radicals is an enzymatic process.

- The AKO relations from the RadicalScavengingEnzyme to ActiveEnzyme in the process frame IrreversibleEnzymeDenaturation represent that the enzyme in the radical scavenging process may denaturate irreversibly to an inactive form.

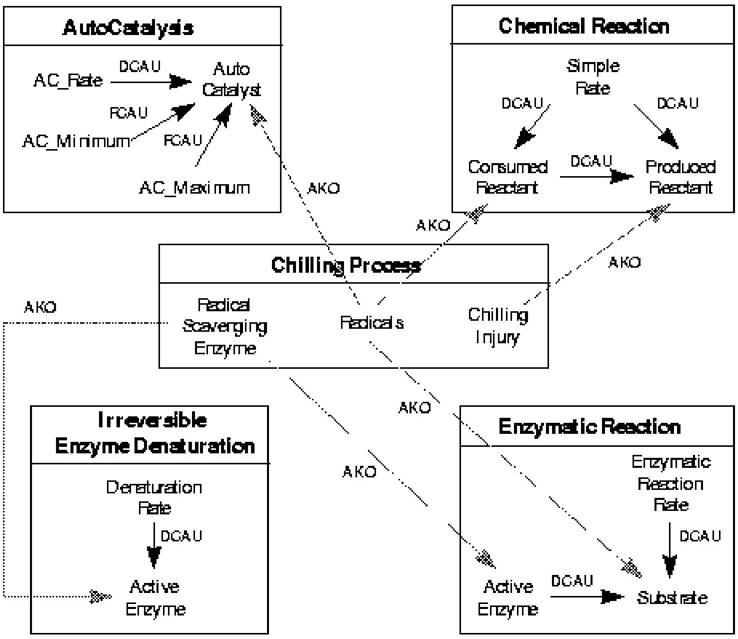


Figure 7.11: Knowledge graph showing how the occurrence of chilling injury is modelled as interactions between generic processes.

The occurrence of chilling injury is a quality attribute that is the visible effect of the amount of free radicals on cell membranes. This is modelled as a behavioural relationship between Radicals and ChillingInjury. This example again shows that quality attributes can be treated as product properties.

7.6.3.2 Construction of the process structure graph

To illustrate the construction of the process structure graph for the occurrence of chilling injury, suppose that the modelling question specifies ChillingInjury as the quantity of interest, and Temperature as the exogenous quantity.

Figure 7.12 (A) displays the initial process structure graph consisting of the quantities in the modelling question. In the first step, displayed in Figure 7.12 (B), the process frame ChillingProcess is included, as this is the only process frame in the DESIMAL library that contains the quantity of interest

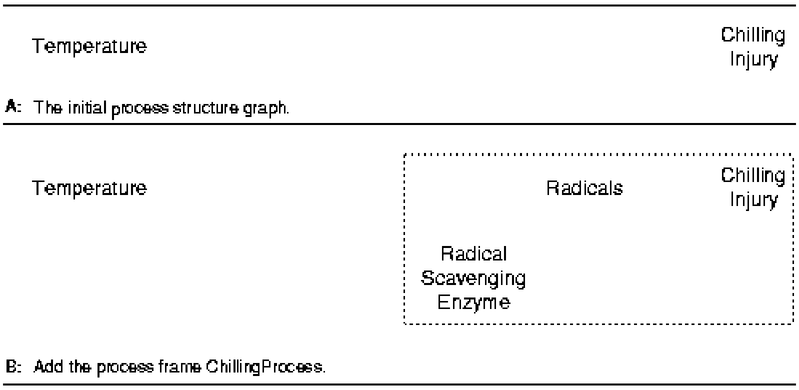
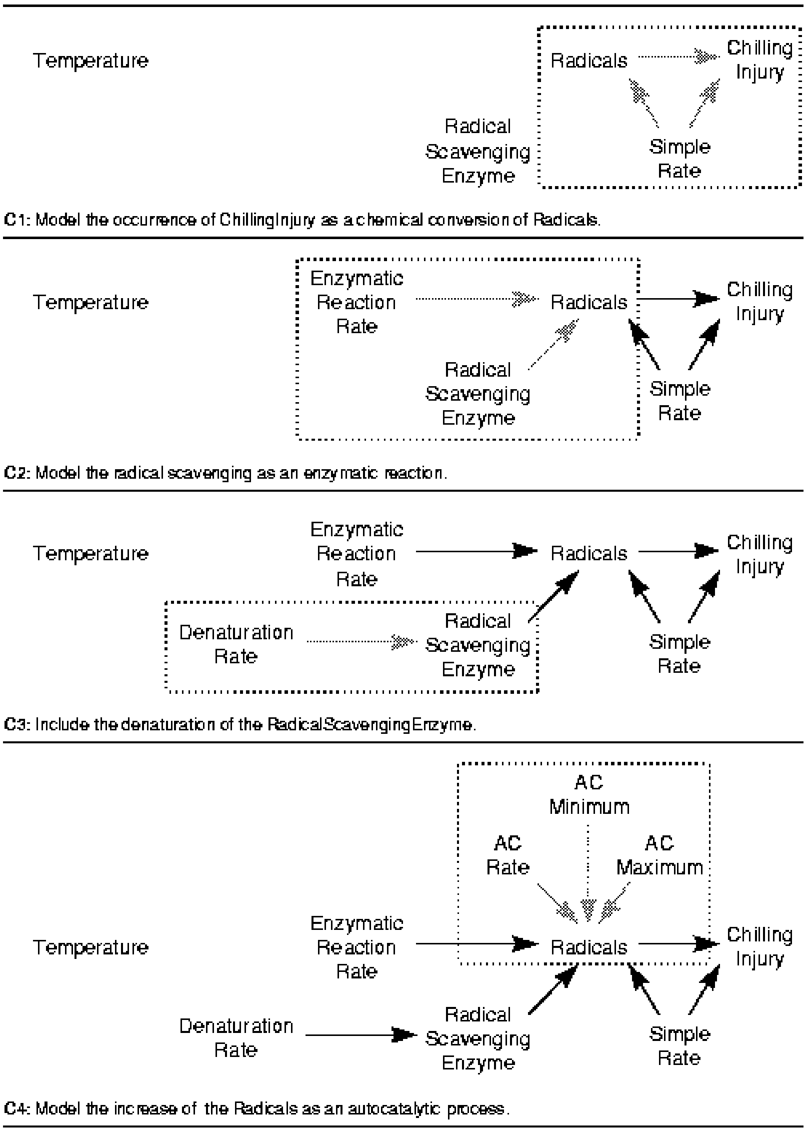

Temperature
Chilling Injury
A: The initial process structure graph.
Temperature
Radicals
Chilling Injury
Radical Scavenging Enzyme
B: Add the process frame ChillingProcess.

Figure 7.12: The first steps in constructing a process structure graph for chilling injury. Graph (A) contains quantities in the modelling question. In (B) the process frame ChillingProcess is added which relates chilling injury to (free) radicals and radical scavenging.

ChillingInjury.

Next, the AKO relations from this process frame are followed, which causes instances of the process frames ChemicalReaction, EnzymaticReaction, IrreversibleEnzymeDenaturation, and AutoCatalysis to be created and included in the process structure graph. This is displayed in Figure 7.13 (C1) through (C4). The process frame ChemicalReaction only contains quantities and EQU relations to the process frames DecayReaction and FormationReaction that specify the relations between these quantities (see Figure 5.5). These process frames are instantiated and included in the process structure graph. This is not displayed.

Finally, instances of the process frame ReactionKinetics are created for SimpleRate, DenaturationRate, and EnzymaticReactionRate, representing these rates depend on Temperature according to Arrhenius' law. For the quantity AC_Rate an instance of the process frame ReactionSetKinetics is created, representing that this quantity is a combination of several chemical rates. The resulting process structure graph is displayed in Figure 7.14.



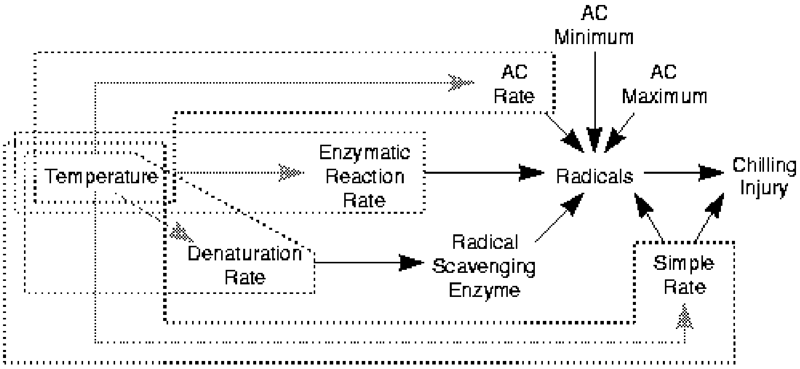
C1: Model the occurrence of ChillingInjury as a chemical conversion of Radicals.

C2: Model the radical scavenging as an enzymatic reaction.

C3: Include the denaturation of the RadicalScavengingEnzyme.

C4: Model the increase of the Radicals as an autocatalytic process.

Figure 7.13: The process frame **ChillingProcess** contains several AKO relations to quantities in four other frames. These frames are added one by one.



D: All reaction rates depend on Temperature.

Figure 7.14: The final process structure graph is obtained by creating instances of the process frame **ReactionKinetics** for all reaction rates in the process structure graph.

Chapter 8

Discussion

8.1 Introduction

This dissertation describes the development of the DESIMAL method for modelling change of agricultural products. The first step was to develop a definition of quality and quality change of agricultural products. This resulted in the conceptual model and the definition of a quality change model presented in Chapter 3.

The second step was to identify the tasks in supporting the construction of such quality change models. The conceptual model shows that the quality change of an agricultural product can always be traced back to a combination of physiological processes occurring in the product. As it is not always obvious which processes are responsible for the observed quality change, the first task in the DESIMAL method is to propose several process decompositions for a quality change phenomenon under study. This Qualitative Process Analysis task was described in Chapter 5. Once the appropriate decomposition into generic processes is found, it is relatively simple to construct a simulation model for the quality change phenomenon. This second task is called Simulation Model Construction task and was described in Chapter 6.

Qualitative Process Analysis and Simulation Model Construction are separate tasks that automatedly construct a model for the quality change phenomenon under study. In Section 8.3 the DESIMAL approach will be compared with other approaches to automated modelling. In the next section a resume of the contributions of this research to the field of modelling quality change of agricultural products is given.

8.2 Main achievements

- A *conceptual model for quality and quality change of agricultural products* was developed, and the concepts of *assigned quality* and *acceptability* of a product were introduced. The assigned quality is a combination of intrinsic product properties and quality criteria imposed on one product in isolation. Acceptability is the assigned quality in relation with other products. We showed that for modelling the effects of the conditions during postharvest distribution on the quality change of the distributed agricultural product only the assigned quality of the agricultural product is important.

- In the many examples of models for physiological phenomena described in this dissertation and elsewhere (e.g. [Tijskens and Sloof, 1996]), the advantages of a strict decomposition of the complex phenomenon into primitive physiological processes were shown. It is this strict decomposition that enables the automated construction of the dynamic product models.

- A graph-based representation language for qualitative modelling has been developed, based on a variant of conceptual graphs, called knowledge graphs [James, 1991]. The language defines a restricted number of relation types: DCAU and FCAU relations to represent behavioural knowledge, PAR relations to represent decompositions, and AKO and EQU relations to represent modelling assumptions. Relations between quantities are grouped into frames to represent physiological processes and decompositions of aggregate quantities. These frames are the building blocks for the models constructed in the Qualitative Process Analysis task.

- Our approach to simulation model construction strictly distinguishes between qualitative and quantitative models. The qualitative model is used for decomposition and finding relevant processes, but not for simulation. This is deliberate and in our opinion justified. For model construction and reuse one needs to abstract from details (both in reality and in implementation) in order to focus on the associations between the models. For simulation, however, quantitative models and precision are needed.

- To connect the qualitative knowledge frames and the quantitative models, an explicit level of applicability knowledge has been introduced. Each application frame specifies a mapping between one quantitative model and a connected graph consisting of one or more knowledge graph frames. In this way, the models at the qualitative level can be developed independent of the models at the quantitative level, and vice versa.
- The DESIMAL system constructs simulation models that have an explicit control component. This enables to combine several simulation components into one large model. It is possible to combine a 'fast' model for stable portions of a process with more precise models for portions that are outside the scope of the 'fast' models. In this way, more efficient simulation models are obtained, without sacrificing the accuracy of the simulation model.
- A working method has been described for constructing executable simulation models for physiological behaviour using a commercially available simulation environment.

8.3 Comparison

In this section, the DESIMAL approach is compared with other approaches to modelling support and automated model construction. Below, the main differences and similarities are summarized between the DESIMAL approach and the four approaches to modelling support and automated model construction that have been described in Chapter 4. In the subsequent sections for each approach we describe where and why the DESIMAL approach differs from that approach.

8.3.1 Differences and similarities between the approaches

The DESIMAL approach, the evolutionary modelling approach of [Top, 1993], the compositional modelling approach of [Falkenhainer and Forbus, 1991], the automated modelling for simulation approach of [Iwasaki and Levy, 1994] and the TRIPEL approach of [Rickel and Porter, 1994] are compared on the following aspects:

- goal of the methods,

- purpose of the constructed models,
- structure of the simulation models,
- inputs to the methods,
- separate description levels,
- representation of the qualitative level,
- representation of the quantitative level,
- determination of the relevant system.

The goal of both the DESIMAL and the evolutionary modelling method of Top is to provide *modelling support*. The method of Iwasaki and Levy, the TRIPEL system and the compositional modelling method of Falkenhainer and Forbus aim at *automated model construction*. The goal of these methods is to construct the simplest adequate model that can explain the behaviour of the system under study.

The models constructed by TRIPEL and by the compositional modelling method of Falkenhainer and Forbus are intended to explain the *immediate behaviour* of a system under study, in reaction to a disturbance. The constructed model has to satisfy only the initial state specified by the question. In contrast, the models constructed by DESIMAL, by the evolutionary modelling method of Top and by the method of Iwasaki and Levy are intended for *simulation* of the system under study. These models describe the behaviour of the system under study over a period of time, and have to satisfy all states that can be reached from the initial state specified in the modelling question.

The simulation models constructed by DESIMAL, by the evolutionary modelling method of Top and by the method of Iwasaki and Levy construct simulation models that consist of a number of submodels. In the simulation models constructed by the evolutionary modelling method of Top the submodels are active throughout a simulation run, and it is not possible to select alternative models during a simulation run. The simulation models that are constructed by DESIMAL and the method of Iwasaki and Levy are collections of submodels that are activated dynamically during a simulation run. For each process or component in the system the simulation model may contain a number of submodels that describe the behaviour of the process or component in different operating regions.

Automated modelling starts from a modelling question and a specification of the system under study. The modelling question specifies the quantities that have to be described by the model. The specification of the system under study contains additional information, such as the physical layout of the components in the system. The methods of Iwasaki and Levy and of Falkenhainer and Forbus both use a separate scenario description. Furthermore, the modelling question used in the method of Iwasaki and Levy contains a set of conditions describing the set of states for which the model is constructed. In the evolutionary modelling method of Top it is assumed that the modeller uses a diagram or a mental model of the system under study to guide the modelling process by specifying the physical layout of the system under study. The TRIPEL and DESIMAL systems only use a modelling question. The system under study is always a subset of the physiological processes occurring in the agricultural product. As the relationships between the processes cannot be manipulated, TRIPEL and DESIMAL do not need an explicit specification of the system under study.

The methods use different description levels to represent the domain knowledge. A description level corresponds with a level of building blocks provided by the method. The description levels are used to bridge the gap between the modeller's knowledge about the system under study and the simulation models. The DESIMAL method defines a qualitative and a quantitative description level. The evolutionary modelling method defines four description levels: functional components, qualitative processes, mathematical relations, and model data. The methods for automated model construction use one description level. Each model fragment at this description level combines several types of knowledge. For example, a model fragment specification in the compositional modelling method of Falkenhainer and Forbus consists of a specification of the structural layout of the modelled concept, relations describing the behaviour of the concept, a specification of the operating region in which the relations are valid, and the assumptions underlying the formulation of the relations. In the methods for modelling support these types of knowledge are represented in separate description levels.

The number of description levels corresponds with the number of modelling tasks defined in the method. Hence, the approaches to automated model construction implement one modelling cycle, while the approaches to modelling support use a number of modelling cycles, one for each description level. When multiple description levels are used, explicit relations have to be defined between the description levels. These relations specify how a model at one description level can be implemented using models at the next description level. In the DESIMAL approach these relations are represented by application frames.

Formal representations are used for the quantitative level and for the description level immediately above the quantitative level. In the Desimal method and the evolutionary modelling method the latter description level represents processes occurring in the domain under study. In Desimal the physiological processes are represented by knowledge graphs. Top uses bond graphs to represent the physical processes.

The specification of the quantitative levels can usually immediately be applied in an executable version of the constructed model. Falkenhainer and Forbus use ready-to-use LISP code to specify the quantitative relations in the model fragments. The models in the Desimal library are specified as variable assignments, that are easily translated into a Prosim model. Top uses mathematical equations that are to be converted into assignments to variables.

The methods use different definitions to determine the part of the system that is relevant for answering the modelling question. These definitions were discussed in detail in Section 5.4.3.

8.3.2 Evolutionary modelling

The evolutionary modelling method uses a structure for a model library, in which each model consists of four description levels [Top, 1993; Top *et al.*, 1995a]. The highest description level consists of functional components, which are building blocks for a functional decomposition of the system for which the model is developed. The next description level represents the physical processes that occur in the functional components. The third description level specifies the mathematical relations for the physical processes. The fourth and lowest description level defines values for the parameters of the mathematical relations.

The physical processes are represented by bond graphs. The nodes in a bond graph represent physical mechanisms such as storage, dissipation, conversion and distribution of energy. The edges represent energy flows described in a complementary pair of physical variables, called flow and effort. The bond graph representation is domain-independent. In each domain, domain-specific concepts can be found for the effort and flow variables. For example, in thermodynamics these variables are temperature and entropy flow, respectively. With bond graphs it is possible to incorporate different domains into one model.

Each bond graph element defines the global form of the mathematical model to be used for the mechanism. The actual form of the mathematical model is found by making additional modelling assumptions.

The use of bond graphs to restrict the possible mathematical models for a system under study corresponds to the use of the knowledge graphs in the DESIMAL approach. For each knowledge graph frame only a restricted set of mathematical models is applicable. The application frames in the DESIMAL library represent knowledge about which mathematical models are applicable for a qualitative knowledge graph frame. As each bond graph element defines the global form of the mathematical relations to be applied for the element, the model library used by Top does not use explicit applicability knowledge.

Compared with the DESIMAL library, the model library used by Top has an additional level of functional components. This description level is not included in DESIMAL, because the constructed quality change models describe the quality change of one agricultural product, and the product is treated as a homogeneous object with respect to the processes occurring in the product.

If processes were allowed to have a different behaviour in different parts of the product, then a component level could be introduced, in which the components correspond to anatomical parts of the product. For example, for a flower such components could be the stem, the leaves, and the petals. The knowledge graph for the process of bacteria growth displayed in Figure 5.4 in fact contains two components: the stem and the vase water. These components are modelled by using separate specialisations of the process frame BacteriaGrowth, involving separate quantities for the bacteria concentrations in the stem and in the vase water. In this example, the vase water is not part of the flower, but belongs to the environment of the flower, so that the bacteria growth in the vase water could also be modelled in the environment model instead of the dynamic product model.

A quality change model constructed by DESIMAL can be used to describe the quality change of a batch of products, provided that the batch contains only products of one cultivar, and that the environmental conditions across the batch are the same or can be modelled as such. If the batch contains different products, for example in case of a mixed load transport [Vogels and Janssens, 1993], or if the environmental conditions may vary across the batch, then the products in the batch have to be divided into groups of products of the same cultivar with homogeneous environmental conditions, and separate dynamic product models have to be used to describe the quality change of the product groups. These dynamic product models will share one environment model for the environmental conditions inside the batch. In this way, the interactions between the products in the batch are correctly modelled by influences on the common environment. In our view, the product groups cannot be modelled as components, as in that case the products would influence each other immediately rather than through the common environment.

In the evolutionary modelling approach the decomposition and the selection of appropriate processes are distinct steps. In the DESIMAL approach the decomposition of a quality change phenomenon into processes and the selection of the appropriate generic processes for the specific processes in the phenomenon are treated as one step. Each process frame in the DESIMAL library represents one physiological process or a well-defined aspect of a physiological process. The behaviour of the process is represented by causal relations either in the process frame itself, or in generic process frames of which the process frame is a specialisation. A composite process is represented by a graph consisting of interacting process frames, and possibly decomposition frames. A decomposition frame represents that one aggregate quantity is the composite of two or more other quantities. The complex process influencing the aggregate quantity is represented by the decomposition frame and the process frames for the processes that affect the constituent quantities.

8.3.3 Iwasaki and Levy

Both the DESIMAL method and the method of Iwasaki and Levy construct models for simulation. The algorithm used by Iwasaki and Levy alternates between determining relevant phenomena and selecting appropriate model fragments. Firstly, the relevant processes are determined by backward chaining on the causal influences on the quantities of interest. The same mechanism is applied in the Qualitative Process Analysis task of DESIMAL. Secondly, for each relevant process one appropriate model fragment is determined by reasoning about the underlying modelling assumptions of the model fragments. Each selected model fragment introduces new quantities for which the influencing processes have to be determined. Selecting an appropriate model fragment corresponds with selecting whether or not to include a decomposition frame in Qualitative Process Analysis. Hence, Iwasaki and Levy always pursue one way of modelling a process, whereas in DESIMAL all ways of modelling a process are pursued, resulting in multiple process structure graphs for the phenomenon under study.

Similar to the DESIMAL method, the models constructed by Iwasaki and Levy consist of a collection of model fragments. At each state during a simulation, from these model fragments a subset is selected that describes the behaviour in that state, This set of model fragments is selected by evaluating conditions of the model fragments, that specify when the process occurs, which objects must exist, and the constraints the objects must satisfy. The selection rules are implicitly coded in the model construction and simulation algorithm. The simulation models constructed by the DESIMAL method have an explicit simulation control component to specify the active models for each simulation state. This explicit simulation control component allows to specify a strategy when more than one model applies to an active process. Examples of such strategies are to apply the simplest model, or to apply the model with the broadest operating range. The method of Iwasaki and Levy by default activates the simplest applicable model fragment.

8.3.4 TRIPEL

The TRIPEL system [Rickel and Porter, 1994; Rickel, 1995] uses one comprehensive, detailed model of the plant physiology domain. The goal of the model construction method is to select a part of this comprehensive model that is relevant for answering a modelling question. The modeller can guide the model construction by specifying quantities to be treated as 'glass-box' quantities or 'black-box' quantities. A 'glass-box' quantity has to be described in terms of subquantities. A 'black-box' quantity has to be described as an aggregate quantity, disregarding its subquantities. Apart from this it is not possible to choose between different models for one process.

In contrast, the DESIMAL library contains models for generic processes that are reused in models for specific phenomena. For one phenomenon a number of process decompositions may be constructed, depending on which decomposition frames are included and depending on the AKO relations between the knowledge graph frames. Hence, the DESIMAL system configures a model by identifying the relevant processes and selecting the appropriate models for these processes.

In the TRIPEL system the building blocks are single causal relations (called influences) between domain quantities. Which influences on a quantity are included in the model depends on the time-scale that is selected by the TRIPEL system. Only influences with a time-scale equal or smaller than the selected time-scale are included. In DESIMAL the building blocks are physiological processes. One process may contain a number of influences on one or more quantities. The boundaries of the processes are determined when developing the DESIMAL library and cannot be changed during model construction. In the TRIPEL system the boundaries of the processes depend on the selected time-scale.

8.3.5 Compositional modelling

The model fragments in the compositional modelling approach of [Falkenhainer and Forbus, 1991] combine both quantitative and qualitative knowledge about a domain concept. In the DESIMAL library these two types of knowledge are separated. The knowledge graph frames library represent only qualitative knowledge, whereas the mathematical models only contain quantitative knowledge. The separation between processes and models enables the construction of a conceptual model of the behaviour of a phenomenon under study, separate from the construction of a mathematical model of the phenomenon. In this way, the DESIMAL system may point the modeller to gaps in the model library. If for a different product the process has been modelled mathematically, the DESIMAL system may propose to use that model.

Falkenhainer and Forbus use three types of modelling assumptions. The first type are *ontological assumptions*, that specify the ontology used to describe the system or phenomenon. For each system, one ontology or a consistent set of ontologies has to be assumed. The second type are *grain assumptions*, that specify which quantities have to be modelled as single, aggregate quantities. These quantities must not be decomposed into their constituent subquantities. The grain assumptions define the level of detail that is used to describe the system under study. The third type are *approximations and abstractions*, that specify the assumptions on which a model fragment is based. Approximations may correspond to ignoring an influence (e.g. ignoring the denaturation of an enzyme), or may represent an assumption about the structure of the environment (e.g. the substrate can be modelled as an infinite source). Abstractions reduce the level of detail that is used to describe a quantity, e.g. by specifying that a quantity has a discrete value instead of a continuous value.

In DESIMAL a modelling assumption is expressed by an AKO relation, representing that a quantity is modelled as a specialisation of a more generic quantity. This corresponds with the abovementioned second type of approximation. The first type of approaximation, ignoring an influence, is not represented as an underlying assumption of a model, but is part of the (refined) modelling question. In our view an influence can only be ignored once all influences in the phenomenon under study are known and understood. As a result, the DESIMAL system first creates a comprehensive model of the phenomenon under study. Subsequently, the modeller may decide which influences can be safely ignored.

Bibliography

[Addanki *et al.*, 1991] S. Addanki, R. Cremonini, and J. S. Penberthy. Graphs of Models. *Artificial Intelligence*, 51:145–177, 1991.

[Böckenhoff and Hamm, 1983] E. Böckenhoff and U. Hamm. Perspektiven des Marktes für Alternativ erzeugte Nahrungsmittel. *Berichteüber Landwirtschaft*, 61:345–381, 1983.

[Chang, 1981] R. Chang. *Physical Chemistry with Applications to Biological Systems.* MacMillan Pub. Co., New York, 1981.

[de Kleer and Brown, 1984] J. de Kleer and J.S. Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83, 1984.

[Falkenhainer and Forbus, 1991] B. Falkenhainer and K.D. Forbus. Compositional modeling: finding the right model for the job. *Artificial Intelligence*, 51:95–143, 1991.

[Falkenhainer and Forbus, 1992] B. Falkenhainer and K.D. Forbus. Composing Task-Specific Models. In *DSC-Vol. 41, Automated Modeling, ASME 1992*, pages 1–9, 1992.

[Forbus, 1984] K. D. Forbus. Qualitative Process Theory. *Artificial Intelligence*, 24:85–168, 1984.

[Forbus, 1990] K.D. Forbus. Qualitative Physics: Past, Present, and Future. In D.S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 11–39. Morgan Kaufmann Publishers, inc., 1990.

[Garvin, 1984] D. A. Garvin. What Does "Product Quality" Really Mean ? *Sloan Management Review*, pages 25–43, Fall 1984.

[Grierson and Kader, 1986] D. Grierson and A.A. Kader. Fruit ripening and quality. In J.G. Atherton and J. Rudich, editors, *The Tomato Crop*, pages 241–275. Chapman and Hall, London, 1986.

[Gruber, 1993] T. R. Gruber. Model Formulation as a Problem-Solving Task: Computer-Assisted Engineering Modeling. *International Journal of Intelligent Systems*, 8(3):105–127, 1993. Special volume on Knowledge acquisition as modeling.

[Iwasaki and Levy, 1994] Y. Iwasaki and A. Y. Levy. Automated Model Selection for Simulation. In *Prceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94*, Seattle, Washington, July 31 – August 4, 1994.

[James, 1991] P. James. Structuring Knowledge using Knowledge Graphs. In *Proceedings of the 5th European Knowledge Acquisition for Knowledge-based Systems Workshop (*EKAW 91*)*, Strathclyde University, Scotland, 1991.

[Juran *et al.*, 1974] J.M. Juran, F.M. Gryna, and R.S. Bingham, editors. *Quality Control Handbook*. McGraw-Hill, New York, 1974.

[Kader *et al.*, 1989] A. A. Kader, D. Zagory, and E. L. Kerbel. Modified atmosphere packaging of fruits and vegetables. *Critical Reviews in Food Science and Nutrition*, 28(1):1–30, 1989.

[Kramer and Twigg, 1983] A. Kramer and B. A. Twigg. *Quality control in the food industry*. Avi, Westport, 3rd edition, 1983.

[Lancaster, 1971] K.J. Lancaster. *Consumer demand: a new approach*. Columbia University Press, New York, 1971.

[Mili, 1995] F. Mili. User Oriented Library Documentation. In *Proceedings of first international workshop on Knowledge-Based systems for the (re)Use of Program Libraries, KBUP'95*, pages 11–20, Sophia-Antipolis, France, November 23-24, 1995.

[Murray and Sheppard, 1988] K. Murray and S. Sheppard. Knowledge-based simulation model specification. *Simulation*, 50(3):112–119, March 1988.

[Nayak, 1995] P. Pandurang Nayak. *Automated Modeling of Physical Systems*. Lecture Notes in Computer Science, vol. 1003. Springer, 1995.

[Olson and Jacoby, 1972] J. C. Olson and J. Jacoby. Cue utilization of the quality perception process. In M. Vekatesan, editor, *Proceedings of the third Annual Conference of the Association for Consumer Research*, pages 167–179, Iowa City, 1972. Association for Consumer Research.

[Penix and Alexander, 1995] J. Penix and P. Alexander. Design Representation for Automating Software Component Reuse. In *Proceedings of first international workshop on Knowledge-Based systems for the (re)Use of Program Libraries, KBUP'95*, pages 75–84, Sophia-Antipolis, France, November 23-24, 1995.

[Rickel and Porter, 1994] J. Rickel and B. Porter. Automated Modeling for Answering Prediction Questions: Selecting the Time Scale and System Boundary. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 1191–1198, Seattle, Washington, July 31 – August 4, 1994.

[Rickel, 1995] J. Rickel. *Automated Modeling of Complex Systems to Answer Prediction Questions*. PhD thesis, University of Texas at Austin, 1995.

[Schoorl and Holt, 1985] D. Schoorl and J.E. Holt. A Methodology for the Management of Quality in Horticultural Distribution. *Agricultural Systems*, 16:199–216, 1985.

[Schut and Bredeweg, 1996] C. Schut and B. Bredeweg. An overview of approaches to qualitative model construction. *The Knowledge Engineering Review*, 11(1):1–25, 1996.

[Shewfelt *et al.*, 1988] R.L. Shewfelt, C.N. Thai, and J.W. Davis. Prediction of changes in colour of tomatoes during ripening at constant temperatures. *Journal of Food Science*, 53:1433–1437, 1988.

[Sloof and Simons, 1993] M. Sloof and A.E. Simons. Knowledge based construction of models to predict cut flower quality. In A. Pave, editor, *Modelling and Simulation, Proceedings of the European Simulation Multiconference, ESM'93*, pages 13–17, Lyon, France, June 7-9, 1993. The Society for Computer Simulation. ISBN 1-56555-056-0.

[Sloof and Simons, 1994] M. Sloof and A.E. Simons. Towards a Task Model for the Design of Simulation Models. In A. Guasch and R.M. Huber, editors, *Proceedings of the Modelling and Simulation Conference, ESM'94*, pages 721–725, Barcelona, Spain, June 1-3, 1994. The Society for Computer Simulation. ISBN 1-56555-028-5.

[Sloof and Tijskens, 1995] M. Sloof and L.M.M. Tijskens. Problem decomposition: Application in experimental research, statistical analysis and modelling. In *Advances in Intelligent Data Analysis volume 1, Proceedings of the IDA-95 symposium*, pages 186–190, Baden-Baden, Germany, August 17-19, 1995.

[Sloof and Willems, 1995] M. Sloof and M. Willems. Automated Modelling Support: Identifying Relevant Aspects of a Problem. In *Proceedings of first international workshop on Knowledge-Based systems for the (re)Use of Program Libraries, KBUP'95*, pages 21–29, Sophia-Antipolis, France, November 23-24, 1995.

[Sloof *et al.*, 1996] M. Sloof, L.M.M. Tijskens, and E.C. Wilkinson. Concepts for Modelling Quality of Perishable Products. *Trends in Food Science and Technology*, 7:165–171, May 1996.

[Sloof, 1998] M. Sloof. Automated Modelling of Physiological Processes During Postharvest Distribution of Agricultural Products. *Artificial Intelligence Review, Special volume on Artificial Intelligence for Biology and Agriculture*, 12(1–3):39–70, 1998.

[Sowa, 1984] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley, Reading, MA, 1984.

[Steenkamp, 1987] J-B. E. M. Steenkamp. Conjoint Measurement in Ham Quality Evaluation. *Journal of Agricultural Economics*, 38:473–480, 1987.

[Steenkamp, 1989] J-B. E. M. Steenkamp. *Product quality: an investigation into the concept and how it is perceived by consumers.* PhD thesis, Agricultural University Wageningen, The Netherlands, 1989.

[Tijskens and Evelo, 1994] L.M.M. Tijskens and R.G. Evelo. Modelling colour of tomatoes during post-harvest storage. *Postharvest Biology and Technology*, 4:85–98, 1994.

[Tijskens and Polderdijk, 1996] L. M. M. Tijskens and J. J. Polderdijk. A generic model on keeping quality of vegetable produce during storage and distribution. *Agricultural Systems*, 51(4):431–452, 1996.

[Tijskens and Sloof, 1996] L.M.M. Tijskens and M. Sloof. How to build models in plant physiology: problem decomposition, fundamental processes and quality attributes. Technical Report 474, ATO-DLO, P.O. box 17, 6700 AA Wageningen, The Netherlands, April - May 1996.

[Tijskens *et al.*, 1994] L.M.M. Tijskens, E.C. Otma, and O. van Kooten. Photosystem II quantum yield as a measure of radical scavengers in chilling injury in cucumber fruits and bell peppers: A static, dynamic and statistical model. *Planta*, 194:478–486, 1994.

[Tijskens *et al.*, 1996] L.M.M. Tijskens, M. Sloof, E.C. Wilkinson, and W.G. van Doorn. A model of the effects of temperature and time on the acceptability of potted plants stored in darkness. *Postharvest Biology and Technology*, 8:293–305, 1996.

[Tijskens, 1995] L. M. M. Tijskens. A Generic Model on Keeping Quality of Horticultural Products, including Influences of Temperature, Initial Quality and Quality Acceptance Limits. In *Proceedings of the 19th International Congress of Refrigeration volume 2*, pages 361–368, The Hague, The Netherlands, August 20–25, 1995. International Institute of Refrigeration (IIR/IIF), Paris, France.

[Top *et al.*, 1995a] J.L. Top, A.P.J. Breunese, J.F. Broenink, and J.M. Akkermans. Structure and use of a library for physical systems models. In *Proceedings ICBGM'95*, pages 97–102, Las Vegas, 1995.

[Top *et al.*, 1995b] J.L. Top, M. Sloof, and J.M. Akkermans. Reusability of Physiologic Simulation Models. In A.J. Udink ten Cate, R. Martin-Clouaire, A.A. Dijkhuizen, and C. Lokhorst, editors, *Proceedings of the 2nd IFAC Workshop on AI in Agriculture*, pages 347–351, Wageningen, The Netherlands, May 29-31, 1995.

[Top, 1993] J.L. Top. *Conceptual modelling of physical systems*. PhD dissertation, University of Twente, Enschede, September 1993.

[van der Sman *et al.*, 1996] R. G. M. van der Sman, R. G. Evelo, E. C. Wilkinson, and W. G. van Doorn. Quality Loss in Packed Rose Flowers due to *Botrytis cinerea* as Related to Temperature Regimes and Package Design. *Postharvest Biology and Technology*, 7(4):341–350, 1996.

[van Doorn and Tijskens, 1991] W. G. van Doorn and L. M. M. Tijskens. FLORES: A Model on the Keeping Quality of Cut Flowers. *Agricultural Systems*, 35:111–127, 1991.

[van Doorn, 1993] W. G. van Doorn. *Vascular occlusion in stems of cut rose flowers*. PhD thesis, Agricultural University, Wageningen, 1993.

[van Langevelde *et al.*, 1992] I.A. van Langevelde, A.W. Philipsen, and J.Treur. Formal Specification of Compositional Architectures. In B.Neumann, editor, *Proceedings of the Tenth European Conference on Artificial Intelligence*, pages 272–276. Wiley & Sons, 1992.

[Vogels and Janssens, 1993] J.W.P.M. Vogels and M.F.M. Janssens. Development of a Decision Support System for Distribution of Mixed Loads. In J. de Baerdemaeker, B. McKenna, M. Janssens, A. Thompson, F. Artés Calero, E. Höhn, and Z. Somogyi, editors, *COST94, Post-Harvest Treatment of Fruit and Vegetables. Proceedings of the workshop on Systems and Operations for Post-Harvest Quality*, Leuven, Belgium, September 14-15 1993.

[Weld and de Kleer, 1990] D. S. Weld and J. de Kleer, editors. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann Publishers, inc., 1990.

[Wilkinson and Polderdijk, 1995] E.C. Wilkinson and J.J. Polderdijk. Q-Bulb: a Model for the Prediction of Tulip Quality. Technical report, ATO-DLO, P.O. Box 17, 6700 AA Wageningen, The Netherlands, January 1995. (in Dutch).

[Wilkinson and Polderdijk, 1996] E.C. Wilkinson and J.J. Polderdijk. Modelling Quality Perception in the Distribution Chain; a Case Study. In J.H. Trienekens and P.J.P.Zuurbier, editors, *Proceedings of the Second International Conference on Chain Management in Agri- and Food Business*, Wageningen, The Netherlands, May 1996.

[Willems, 1993] M. Willems. *Chemistry of Language: a graph-theoretic study of linguistic semantics.* PhD dissertation, University of Twente, Enschede, January 1993.

[Xia and Smith, 1996] S. Xia and N. Smith. Automated modelling: a discussion and review. *The Knowledge Engineering Review*, 11(2):137–160, 1996.

Samenvatting

Fysiologie van modelleren van kwaliteitsverloop
Geautomatiseerd modelleren van kwaliteitsverloop van agrarische produkten

Distributieketens worden zodanig opgesteld dat de gedistribueerde produkten na de produktie zo snel mogelijk bij de consument afgeleverd kunnen worden. Bij de distributie van agrarische produkten speelt de doorlooptijd een nog grotere rol. Omdat agrarische produkten ook na de oogst levende produkten blijven, zal het produkt tussen het moment van oogst en de consumptie veranderen. In het algemeen leidt deze verandering tot een slechter uiterlijk van het produkt en daarmee een slechtere kwaliteit. Een belangrijke taak in de distributie van agrarische produkten is deze achteruitgang in kwaliteit zo veel mogelijk te beperken. Dit gebeurt door de doorlooptijd in de distributieketen zo kort mogelijk te houden en door de omgevingscondities voor de verhandelde produkten te optimaliseren met betrekking tot kwaliteit.

Als hulpmiddel bij de optimalisatie van de omgevingscondities worden kwaliteitsverloopmodellen (KVM) gebruikt. Een KVM is simulatiemodel waarmee de relatie tussen de omgevingscondities en kwaliteitsverloop van een agrarisch produkt bestudeerd kan worden. Totnutoe kostte het opstellen van een kwaliteitsverloopmodel veel tijd en had een kwaliteitsverloopmodel slechts een beperkte toepasbaarheid. Het doel van het onderzoek was een methode te ontwikkelen die de constructie van kwaliteitsverloopmodellen vereenvoudigt.

De ontwikkeling en de bruikbaarheid van een KVM werd sterk beperkt doordat een KVM een rechtstreekse koppeling trachtte te beschrijven tussen de omgevingscondities en de kwaliteit van het produkt. In dit proefschrift is een conceptueel model ontwikkeld voor kwaliteit van agrarische produkten. In dit model wordt benadrukt dat de kwaliteit van een produkt een beoordeling is van de (fysiologische) toestand van het produkt ten opzichte van door de consument gehanteerde kwaliteitseisen. Of een produkt daadwerkelijk gekocht wordt hangt naast de toegekende kwaliteit mede af van marktfactoren, zoals de beschikbaarheid en kwaliteit van andere produkten, het uiterlijk van de verpakking en de service in de winkel.

Voor de evaluatie van de distributieketen voor een produkt kunnen de marktfactoren buiten beschouwing gelaten worden, zodat alleen de reactie van het produkt op de omgevingscondities en de toekenning van kwaliteit aan het produkt van belang zijn. Een kwaliteitsverloopmodel bestaat dan ook uit drie deelmodellen: het eerste deelmodel beschrijft het verloop van de omgevingscondities, het tweede deelmodel beschrijft het verloop in de tijd van eigenschappen van het produkt in relatie tot deze omgevingscondities en wordt dynamisch produkt model genoemd, en het derde deelmodel vertaalt de waarden van deze produkteigenschappen in een maat voor de kwaliteit.

In dit proefschrift is een intelligente modellleermethode ontwikkeld voor geautomatiseerde constructie van dyanmsiche produkt modellen. Het uitgangspunt van de methode is dat het complexe gedrag van de produkteigenschappen het gevolg is van onderlinge interacties tussen primitieve fysiologische processen. Ieder fysiologisch proces in het produkt wordt beïnvloed door omgevingscondities en door andere fysiologische processen.

De constructie van een dynamisch produkt model is opgedeeld in twee aparte modelleertaken.

- *Decompositie van het produktgedrag in primitieve processen.* In de eerste modelleertaak worden technieken voor kwalitatief redeneren toegepast om het produktgedrag te beschrijven als het gevolg van interacties tussen een aantal primitieve fysiologische processen. De bouwstenen in deze modelleertaak zijn kennisgrafen die elk een proces of een decompositie van een produkteigenschap representeren.
- *Constructie van een kwantitatief simulatiemodel.* De tweede modelleertaak stelt een model op voor simulatie van het produktgedrag dat bestaat uit kwantitatieve modellen voor de processen waarin het produktgedrag is opgedeeld. De bouwstenen in de tweede modelleertaak zijn application frames die de kwalitatieve processen koppelen aan de kwantitatieve modellen.

De hoofdstukken 2 en 3 handelen over kwaliteit van agrarische produkten. Hoofdstuk 2 bespreekt literatuur waarin het begrip kwaliteit vanuit verschillende invalshoeken gedefinieerd wordt. Hoofdstuk 3 presenteert het conceptuele model voor kwaliteit van agrarische produkten. Het belangrijkste kenmerk van dit conceptuele model is het onderscheid tussen kwaliteit van één produkt zoals toegekend door een consument, en de accepteerbaarheid van een produkt waarin behalve de toegekende kwaliteit ook marktfactoren meegewogen worden.

De hoofdstukken 4 tot en met 7 handelen over geautomatiseerde constructie van simulatiemodellen voor kwaliteitsverloop van agrarische produkten. Hoofdstuk 4 bespreekt literatuur over automatische model constructie en over ondersteuning van het modelleerproces, en presenteert de globale structuur van de modelleermethode. Hoofdstuk 5 bespreekt de eerste modelleertaak in de voorgestelde methode, waarin het produktgedrag opgedeeld wordt in een aantal primitieve fysiologische processen. Hoofdstuk 6 bespreekt de tweede modelleertaak, waarin een kwantitatief simulatiemodel opgesteld wordt uitgaande van de gevonden proces decompositie en gebruik makend van een collectie kwantitatieve modellen voor de afzonderlijke fysiologische processen. Hoofdstuk 7 bespreekt de Desimal modelleeromgeving en illustreert de modelleermethode aan de hand van een aantal uitgewerkte voorbeelden.

In hoofdstuk 8 wordt de modelleermethode vergeleken met andere methoden voor automatische model constructie en voor ondersteuning van het modelleerproces.