Creating a meta model describing a production system's concepts and their relations

Freek van den Berg s0517593
Bachelor Thesis Information Science
Radboud University Nijmegen, The Netherlands
February 7th, 2009

Abstract:
This thesis has done a literature study about what is expected from production systems to be successful and has found generalization among several theories about this topic. These theories include workflow systems, system theory, model of superior returns and model verification. The result of the literature study has lead to insights that helped with formalizing the concepts and relations that are used in a production system. A meta model in the ORM language has been created iteratively in which the concepts and their relations are expressed as respectively object types and fact types. This iterative creation is done by telling a fictional story about someone cooking in the kitchen and all the activities he engages in and parallel to this creating a meta model. The kitchen setting has been chosen to make it easy for a big group of people to relate to this story. The meta model is generalized after which the finding was that a system is controlled by an entity called control. The responsibility of this control entity is to make a planning, based on the incoming orders and what has happened before in the system. The validity of the model is checked by linking it back to all the literature of the literature study and identifying what is mentioned in the literature in the meta model. Future works deals with how to improve the model, apply it in practice and mathematically reason with it.

Table of contents

Introduction

Nowadays there are many different frameworks, models and theories available on how to manage and describe a system or company. The problem with these frameworks, models and theories is that they are often either very abstract or very informal. Another point they normally lacking at is that they do not connect well to the language of the domain experts which are normally people with limited formal knowledge. What is needed is a framework that offers a description of a system or company in terms that are understandable for as many people as possible but at the same time is formal enough to make it useful in practice.

In addition to that, every business and system seems to have their own terminology, which makes it hard to communicate outside the boundaries of the company or system. For example, people of different companies talking with each other or different systems communicating to each other through a certain protocol tend to assign different meanings to a certain word or description. A taxonomy in which general underlying concepts are shared and defined formally is very wanted here.

The aim of this thesis is to offer a meta model which hopefully can be used to manage a system. The term meta model is used here instead of model to illustrate that the population of the meta model is actually the model of a company. This means that different companies can be modeled using the meta model and that the focus is not on one specific company. Besides creating a meta model, a terminology will be introduced to label the concepts. In order to achieve this, a broad set of different theories have been examined and their key points have been highlighted. From system theory the concept of essential variables is taken, which has been made concrete with knowledge management theory. This resulted in both producing something that's useful for the environment as being able to fulfill the wishes of the environment in time being essential variables. A. Mader et al's model [1] makes the distinction between a concrete and abstract system in order to achieve verification and proof about achieving what one likes to achieve. In order to describe and label the concepts of a system in a general way that any person can understand, something called the kitchen metaphor taxonomy has been introduced. Finally three papers of work flow systems have been investigated to see if the meta model as introduced can satisfy criteria as demanded in this area.

The two problems as mentioned in the first two paragraphs, combined with the theoretical background in the third, lead to an approach in which formalization and usage of the kitchen metaphor terms are the two main factors for creating a meta model for a production system. This leads to the following research question:

How to create a meta model for describing a production system, that is both formal and understandable?

In order to answer the research several steps are taken. After this introduction, this thesis will start with a literature study to see what's already available on designing and managing systems. In the study the terminology of the so called kitchen metaphor is being investigated and what is actually needed for a system or company to survive in its environment.

In the third chapter after this the meta model is iteratively created. This is done by using a fictional story in which John is involved in cooking a meal. This setting has been chosen for to reasons:

- Cooking a meal is an activity many people can relate to and therefore this illustrates well what is going on and makes the story accessible to a big group of people
- The kitchen metaphor uses the terminology as used in the kitchen and therefore makes it easy to make a one on one mapping between the story and the terminology of the kitchen metaphor.

The creation of the meta model is done by using the ORM language (a formal language for defining concepts and their relations) as a meta meta model. For every step John takes in the kitchen, concepts are generated in the meta model simultaneously. Eventually the result of all the steps is merged, resulting into a full conceptual ORM model of a production system environment. In the two paragraphs after this in the same chapter respectively the design decisions are defended and abstraction and composition is addressed.

By linking the created meta model back to the initial literature and showing how the current literature can be placed in the model, the validity of the model is tested after that. After all, when what is going on in the original literature can not be found in the final meta model of this thesis, the meta model is not satisfactory.

After that the meta model is applied on (abstract) problems that occur in a production system, such as:

- Not having enough resources
- The time needed to fulfill an order
- Resources being optimally used

As this thesis will show the creation of the meta model brings up a lot of new questions and a lot of new things to do. A chapter devoted to future work will deal with this and offers suggestions how to continue from here using three categories:

- What is needed to make the meta model operational in practice?
- How can the meta model be improved?
- In what ways can mathematics play a role in reasoning with the meta model?

Finally a conclusion chapter wraps up everything important mentioned in the thesis and the key points it has offered.

Current literature

In order to figure out what is expected from a production system in order to be considered successful, the current literature on this has been examined. Different scientist have proposed different views, models, frameworks and approaches to describe this and the challenge here is to find a common property amongst all of them. In order to achieve this a short summary of several theories is given, combined with linking the view to the viability of a system. The set of theories that have been selected is very broad in order to prevent a bias to a certain view.

System Theory

First Ashby's system theory[2], which has been used for years in both science and society. The first main point Ashby makes is that in designing and setting up a viable system, three stages can be distinguished which he refers to as control, design and regulation. In this context viable can be seen as a system being able to survive which is used as a measure for being successful.

The first stage named control is about detecting the essential variables of a system and the norms they should have. If one takes the system human for example, body temperature can be a very important property for survival and the norm would likely be something like between 36 and 40 degrees Celsius. This is just one variable and using Ashby's theory one can define as many variables as wanted. One thing Ashby doesn't define is how to obtain the variables that are essential. Obtaining them can both be done in a qualitative as a quantitative way. The point he makes about this is that domain experts should be able to know what is important and what the corresponding norms are. This could be seen as a difficulty, but on the other hand it leaves things open for other theories to define what concretely needs to be done here. This which will be shown later in the paragraph.

The second stage in Ashby's model named design, deals with making sure that the essential variables for the system stay within the needed norms. The environment of the system offers treats to the system now and then, which are called disturbances by Ashby. In the concrete example of the previously mentioned human a very bad flu could make the body temperature go up very much and therefore threaten the viability of the system. Ashby proposes that one should design actions to counter these effects called regulations. A regulation for the human case would be to take a lot of rest, drink a lot of water, have a good health insurance and/or take the appropriate medicines. An important thing to note about the design phase is that every single disturbance possible should be detected. Once again Ashby doesn't give an answer about how to achieve this, but also in this case asking a domain expert is a straightforward thing to do.

Finally the third phase, regulation. This phase is about making all the just mentioned happen in practice. This includes creating the right infrastructure to detect disturbances and to execute the right regulation actions. What can be seen from Ashby's theory is that finding the right essential variables and norms is the important thing to do.

Model of superior returns

In the management discipline one often thinks of successful organizations in terms of how successful they are in making money. Normally this money is made by selling products or offering services to the market or environment. There's two views on this concept, namely the I/O model of superior returns and the resource based model. The former model has been introduced by J. Tirole[3], while the later model is often cited and written clearly by M.Peteraf[4].

The I/O model focuses on doing research on the market first. After this figuring out what the market needs is done. The third step involves improving the capabilities of the system in order to be able to create what the market is asking for. Finally the company should offer that what the market asks for and thus rewards with money in return. On the contrary the resource based model focuses on the resources first, then see what resources it can provide for the market and then offering this to the market.

Applying this to Ashby's theory two major essential variables can be detected. From the I/O model, but also implicitly from the resource based model, one can conclude that the demand of the environment for products a company produces is important. When this demand is higher the reward will be greater, inducing a more successful system.

The other variable, derived from both views, is how quickly the company can satisfy this demand. In other words, the average time between and order being placed and an order being processed successfully. When this time is too high two complications occur:

- The environment gets impatient and/or needy and tries to find a different way to satisfy its need. Likely by going to a competing company.
- The product as offered is not needed by the environment anymore. This can for example happen when products are quickly replaced by newer versions through innovation.

Of course one needs to do both things in an efficient way so that eventually there is a profit left after the products are sold, but that will not be the main focus in the thesis. In terms of disturbances two things can go wrong:

- For some reason the market is not interested in your product anymore
- For some reason you can't offer the products needed quickly enough or perhaps not at all.

The first problem can be regulated by making sure you have the right capabilities to offer what the market wants. The mentioned I/O model addresses this after the market has been investigated. This is done by upgrading resources qualitatively (buying new machines, hiring new highly qualified humans, training the current humans etc.), which can be seen as a regulating action.

The second problem about offering products in time is done by either using your resources more efficiently (machines should be running as much as possible, humans should do the most intelligent work they can do instead of wasting time on bureaucracy) or by upgrading resources quantitatively (buying more of the same kind machines, hiring more humans with the same qualities already being there). All the just mentioned can be translated to a diagram (referred to as a regulating matrix by Ashbey) in the following way:

EV \ Disturbance	**Product demand dropping**	**Order time increasing**	**Orders can't be fulfilled**
Product demand	Upgrade resources qualitatively		
Order time		Upgrade resources quantitatively, Use resources more efficiently	Upgrade resources quantitatively, Use resources more efficiently

Since the focus of this thesis is mostly on the internal part of the system, the focus will be on the order time and the product demand will be left out of the picture. This means that the assumption is that what the system will produce is useful for its environment. As can be seen later this is made concrete by orders coming in from the environment without worrying whether the amount of orders that come in is satisfactory.

Verification model for embedded systems

Third the paper 'The construction of Verification Model for embedded systems' of A. Mader, H. Wupper and M. Boon is used[1]. The paper links real systems with abstract formal models. When one makes a model of the real system, one needs to have confidence that the model is expressive enough to describe what is going on in the system and that no matter what the system's behavior will comply with the model.

When this is the case, the model could be used for verifying whether what one wants to achieve can be achieved in the real system. This is done by measuring properties in the real system and use them as input for the model. The framework used for this is expressed in the following square diagram:

Blue print, verification model	Proof	**Formal properties**
Modeling		Formalising
Embedded system, artefact	Confidence	**Behaviour, requirements**

In the framework proposed in this thesis, the model should have both a requirements element as well as an element to model the behavior (read: measurements from the real system). Combining this with the resources and products mentioned in the previous paragraph this results into three categories that need to be managed.

First (labeled modeling in the previous table) there is the formal entities product and instrument (read: resource) that need to interact with each other and be compatible with each other. In terms of the person having flu and high fever, an example of this case would be that a certain treatment has proven to be effective before (theoretical instrument) to create a healthy person again (theoretical product). This is merely a theoretical fact, which can be based on empirical evidence in history or qualitative analyses (such as a doctor's opinion). Note that this theoretical part of the model does not involve time, since what's going on there is considered an objective fact which is always true, independent of any time.

Secondly doing calculations on this model would be the proof step in the previous diagram.

Thirdly (labeled formalization) there is the true treatment performed by a specific doctor at a certain time and a person being sick at that time in real world. The doctor measures a fever before his treatment and rechecks the body temperature after the treatment again. This is the practical half of the model including a specific doctor (practical instrument) and a specific patient (practical product) engaging in interactions at a certain point in time. The important thing here is knowing what and how to measure and act on the environment.

The combination of the three steps result into have confidence in that what one wants to happen or has proven to happen, will happen.

Kitchen metaphor

Fourth to make everything just mentioned easily understandable in a metaphorical way, the kitchen metaphor is introduced. The kitchen metaphor is just a locally used concept at the Radboud University for looking at a production system as if they were kitchens. Therefore, the metaphor has not been found as a result of the literature study, but has addressed due to being part of a course named Introduction Information Architecture in the bachelor program of Information Science.

The definitions are available at the digital restricted Internet environment of the Radboud University in Nijmegen[5], but are also summarized in the table down here. This information has only been used as a starting point and has been extended whenever needed. In addition to that synonyms have been introduced in case different terms are used in the same way but in a different context, such as different professional fields or different theoretical backgrounds. Much of this has been done using common sense. Theoretically when one creates a meta model a set of concepts and their relationships is created and because of that the actual naming of the concepts does not even matter at all since the main goal is to point out the underlying concepts that show up in comparable situations.

The kitchen metaphor describes a production plant in terms of actions that happen in a kitchen. This is useful for two reasons:

- Most if not all people are familiar with what's going on in a kitchen.
- The main purpose of a kitchen is producing a product (namely food) using raw materials (ingredients) which matches a production plant's philosophy very well.

The complete terminology from the course Introduction Information Architecture is a follows:

Term	**World**	**Description in natural language**
product object	Reality	Substances, information, people, or money that are objects of processing
instrument	Reality	A thing that is used for production
production step activity	Reality	A procedure that takes some products as ingredients (precondition), performs some operation, thereby using some instruments for a certain time, and produces some products (postcondition)
batch	Reality	A quantity of products moving from production step to production step resulting in one end product
service	Reality	The iterative execution of a production step with the same instrument for different batches
scenario	Reality	All production steps that actually happen in a period of time
recipe	Description	The specification of a class of batches that result in a certain product
functionality	Description	The specification of a service
order	Description	A bag of pairs (product, due date)

Work flow systems

Finally a link to work flow systems has been made. An area in which the meta model as proposed in this thesis can play a role. Three papers have been selected in order to fulfill this connection [10][11] [12]. In the first and second paper the focus is on what work flows actually are and how they can be modeled and the last paper is a commercial solution as a reply to this in the shape of a language called BPMN (Business Process Modeling Notation). The main point of all papers shall be mentioned here, after which the contents of the paper are used later for validating the model.

The point that is being made in all the papers is that there is some atomic unit of work which is independent of a specific case and called a task. Every task has some input and output and by connecting different tasks (while keeping matching inputs and outputs in mind) a bigger piece of work can be done in a controlled way. As will be shown later this literature can be mapped to the concepts as named in the kitchen metaphor meta model terminology, such as doing small steps in a recipe in order to achieve a big result (a full meal) in the end.

Looking more specifically at the papers, the 2003 paper of W.M.P. Van der Aalst [10] mentions three ways for looking at work flows with respect to control as listed in the following table:

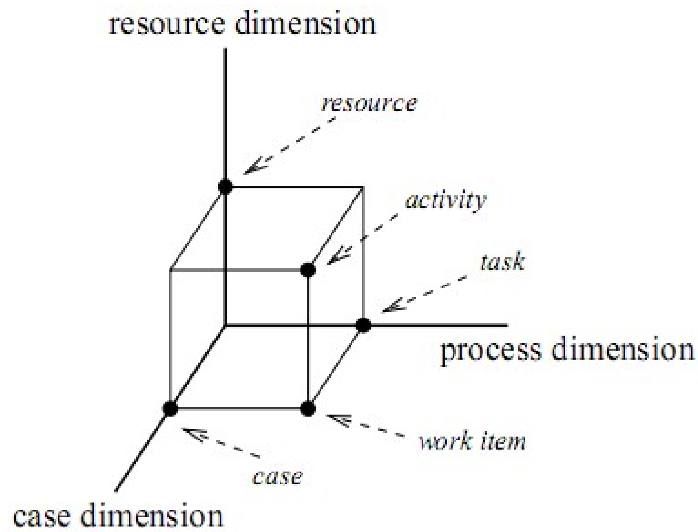
Control	**Description**
Control-flow perspective	Focus activities and their execution ordering. When activities happen in a certain order, the result should turn out right.
Data perspective	Layering business and process data. Every piece of work actually happening needs to have a goal and meet some requirements, which can be found in the business data/abstract world of the company.
Resource perspective	This perspective assigns roles and responsibilities to different people and/or machines in a company and when everyone does their work well, a good result will be achieved.

After the meta model has been created in this thesis, checking for presence of these control perspectives in the model will be one of the steps to validate the model.
Besides that the same paper deals with different kinds of flows one can think of in a work flow besides the traditional AND and OR. Also this aspect is briefly tested in the model validation chapter.

In the 1998 paper of Van der Aalst [11] a link between petri nets and work flows is being made. Petri nets have been around since the early 80's and provide a very formal mechanism and model to look at flow problems. Petri nets consist of places, transitions and arcs (that describe the flow relation). In order to set up the right net, three dimensions have to be taken into account, which are listed in the following table:

Dimension	Description
Resource dimension	Resources that are needed to transform products into something more advanced
Process dimension	The tasks and the routing along these tasks
Case dimension	A specific case that needs to be dealt with

This is enhanced by creating a picture and introducing some new terms by Van der Aalst in the following way:



The three dimensions will come back shortly in the validation of the model of this thesis. The validation is done by checking the ability to address the resource, activity, task, work item and case concept in the meta model as proposed in this thesis. The petri nets will not be taken into account in this thesis, because once the dimensions can be addressed, the paper of Van der Aalst can provide insights about how to map work flows to petri nets.

Finally there is IBM's paper on BPMN, which has been already used in practice a lot and is the standard for specifying business processes in a work flow. This language is maintained by the Object Management group. The main building bricks in this notation are flow objects and connection objects. Flow objects consist of events, gateways and activities:

Flow object	**Description**
Event	An interaction with the outside world, such as a trigger or an impact
Gateway	Used to deal with divergence and convergence in the process flow
Activity	An activity is a detailed description of some work that can be done without having a specific case there at forehand (see: task in the above diagram)

All of the above objects can be connected using connecting objects, which also come in three flavors, namely:

Connecting object	**Description**
Sequence flow	Connecting two objects that needed to be performed in a certain order
Message flow	Connecting two processes that exchange information
Association	Connecting data, text and artifacts with flow objects. Often this kind of line is used for comment of some kind.

In the paragraph about the model validation, the above language with its concepts is applied on the model.

Creating the meta model

This chapter deals with creating a meta model satisfying the two criteria as mentioned previously. These criteria are that the meta model is formal and the meta model is expressed in understandable concepts and terms (which is in this case the kitchen metaphor). After that two paragraphs are dedicated to the design decisions and how to deal with composition and decomposition.

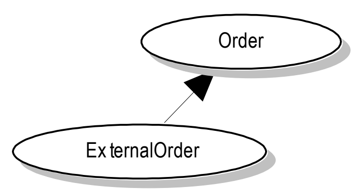
A way to do create a meta model in terms of the kitchen metaphor is to look at a fictional situation of what could happen in a kitchen and detecting formal concepts in it. By distinguishing different steps that are taken in the kitchen while a meal is being prepared and linking these steps to the concepts with terminology of the kitchen metaphor, one can iteratively built a model that is expressed in the wanted language. For the formalization part one has several choices of which ORM has been chosen to be the meta meta model language here because of its strengths in dealing with concepts and relationships between them. Note that the intention here is not to describe perfectly how a meal is prepared. Therefore details have sometimes been omitted due to conceptual irrelevance. For example, not the full list of ingredients is given, but just a few to give an impression of what concepts play a role.

Incoming order

The fictional situation described here, starts with John who wants to surprise his girlfriend with an own made meal for Valentine's day. The scope of this system is the kitchen of John in which resources are available to produce things. This kitchen receives ingredients from the environment (for example doing groceries in the supermarket and carrying them into the kitchen through the door) and hands a meal back to the environment again (for example John putting a nicely cooked meal in the living room).

The intention John has by making a meal can be seen as an order, because John wants to contribute something to the environment. However, although the environment does not explicitly ask for the order it is still considered external, because its result will not be used by the system again. This in contrast to orders of which the resulting product is reused by the system. This introduces the following concepts:

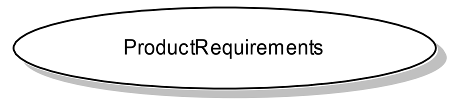
Order (John's Valentine's dinner meal)
ExternalOrder (John's Valentine's dinner meal)



Note that the creation of the meta model does not deal with what a Valentine's dinner meal actually is and how it can be measured in the real world. This step and all future steps in the model creation will just name the concepts as we know them intuitively and in order to make them concrete one needs to link them to the real world somehow. In ORM this is normally done by using labels. Later however, there will follow a conceptual view on the taxonomy being created in natural language. Also synonyms for certain terms will be given, since the system or company view are normally approached differently by people.

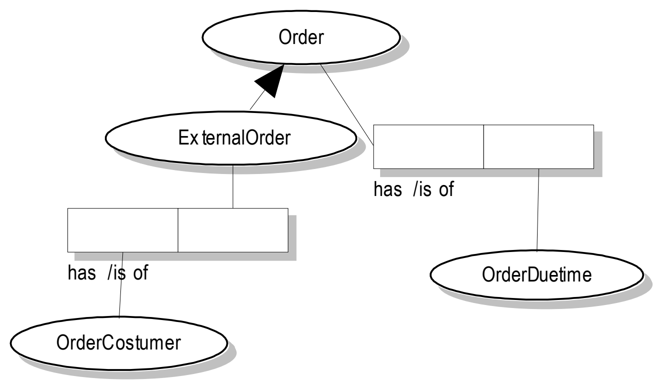
Now John needs to decide what meal he is going to make. In order words, he needs to make requirements for what needs to be achieved to make the order successful. In order to make an order successful a product has to be delivered, but since the concrete product has not been produced yet, it only makes sense to talk about the product requirements now. John decides to make a Scottish Kale oven dish. This leads to:

ProductRequirements(Scottish Kale oven dish)

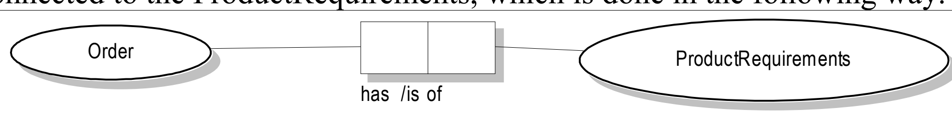


The story continues. John has decided that 18:00 would be a nice time to have the meal ready and therefore there is a requirement to have the meal done by this time. This is the time a definition of an order is being made. An order is a demand/wish for a certain product with ProductRequirements to be delivered by the company or system to a certain person or entity outside the system (or in case of an internal order to the system itself) at a certain time. Despite John's meal being requested by himself, an external entity outside the system is supposed to benefit from it, namely John's girlfriend. This introduces the following concepts:

Costumer(John's girlfriend)
OrderCostumer(John's girlfriend)
OrderDuetime(18:00 at Valentine's day 2009)



Note that OrderCustomer is connected to the ExternalOrder, while OrderDuetime is connected to order. This is so because only external orders have a costumer, while a due time can be assigned to all orders, including internal orders which will be introduced later. The population of the fact types connecting the concepts are omitted, because by knowing the population of the different concepts one can see straightforwardly what can be connected using the story of John mentioned here. In addition to this the order needs to be connected to the ProductRequirements, which is done in the following way:



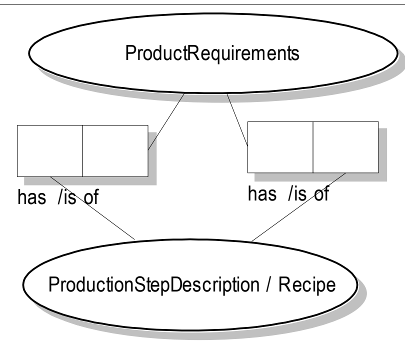
Processing the order

Since John isn't the greatest cook around, he decided to look for a recipe which describes how the meal needs to be prepared. After browsing around on the Internet the following (in Dutch language) webpage has been found:
http://www.receptenweb.nl/SiteNL/servlet/ctl/jsp/RecipeSearch/Recipe.jsp?index=1&word=boerenkool&scope=4&isSMSSearch=false
The recipe is also listed in Appendix A of this thesis including a translation to English.

Thinking more carefully one can say a recipe describes a (relatively big) production step. The term production step description is introduced here to indicate that the recipe is not about a concrete production step happening. After all, the recipe can be reused by more people and is therefore not something only connected to a single case.
It turns out that a recipe (or production step description) contains several parts, namely:

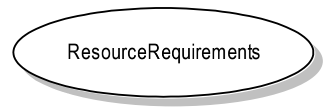
- An ingredient list
- A required resources list
- A description of several steps that need to be taken using the ingredients and resources
- A final product that's promised

Firstly the ingredient list and the promised output product are discussed. Ingredients can be seen as input products for the execution of the recipe to receive a wanted result. The only way the recipe can promise a certain quality of the final product (read: the final product having certain product requirements), is when a precondition stating that the ingredients used fulfill certain product requirements. This means that a recipe (or more general a production step description) is connected twice to ProductRequirements. Once to indicate the input and another time to indicate the output it formally promises to generate. In diagram form this is expressed as follows:

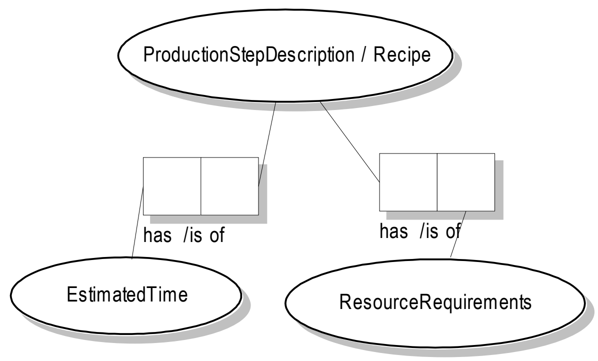
Recipe (How to make Scottish Kale oven dish) ProductRequirements(500 grams of Scottish Kale) ProductRequirements(One piece of smoked saucage) ProductRequirements(Scottish Kale oven dish)	

The slash indicates that both terms can be used interchangeably. Note that not the whole population of ProductRequirements is given here. Just a sample population to get an impression. The 500 grams of Scottish Kale and the saucage are linked to the input role of the recipe, since they are ingredients, while the Scottish Kale oven dish is linked to the output, being the final result when the recipe is executed.

Secondly the required resources list. This list contains all the resources that are needed to create the meal. Some examples in this case are a pan, a oven dish, some spoons and a oven. But this list does not include the most important resource, which is not mentioned in the recipe, but still implicitly assumed to be there. Namely a person to prepare the meal. Therefore this person will also be part of the population of resources. Since a recipe is only an abstract something and still no concrete actions have been taken, the objects that are talked about here are mere the requirements some concrete resources should have. In a similar way as ProductRequirements, ResourceRequirements is therefore introduced:

ResourceRequirements(A human to prepare) ResourceRequirements(oven dish) ResourceRequirements(a pan)	

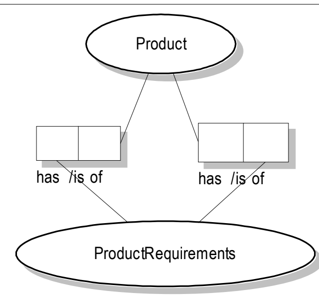
Thirdly and finally the description of the several steps. Every step in the recipe requires a certain input (ProductRequirements and ResourceRequirements) and promises a certain output (ProductRequirements). In addition to this (although not always mentioned in the recipe) an estimated time of this step can be added. This estimated time turns out to be vital for planning later. Note that every step of the recipe is a ProductionStepDescription and therefore a new recipe itself. This can be explained by a recipe being a composition of smaller recipes. Above what has been mentioned about a recipe already, therefore only ResourcesRequirements and EstimatedTime need to be added:

EstimatedTime(1 hour)	

Preparing the order

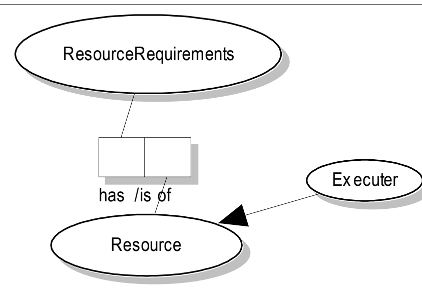
So far John has not done any physical action in preparing the meal. Therefore it is fair to state the everything of the model being generated until now is about the abstract world. Next John doing his concrete physical actions is analyzed. The assumption here is that John intuitively knows what to do to get the meal done at the right time, but after his sequence of actions has been described a reflection is made. This reflection deals with describing how the abstract and concrete world could have been compared.

First of all John goes to the store to buy the ingredients he needs. In principle it does not matter what John buys as long as the products he buys fulfill the product requirements that are needed for the order. This introduces new products that satisfy the requirements that are mentioned before leading to the following addition:

Product (500 gr. AH Scottish kale, the first one on the shelf) Product (1000 gr Mashed patatoes of brand X) Product (1 piece of Unox saucage)	

Note that the concept of the output of the meal has already been added, although John has not cooked anything yet. When John has prepared the meal, the above products can be removed from the population and the result of the cooking can be added. Also the link to productRequirements will then change by having the products above removed through the input fact and (hopefully if the cooking is successful) having the newly made product added through the output fact.

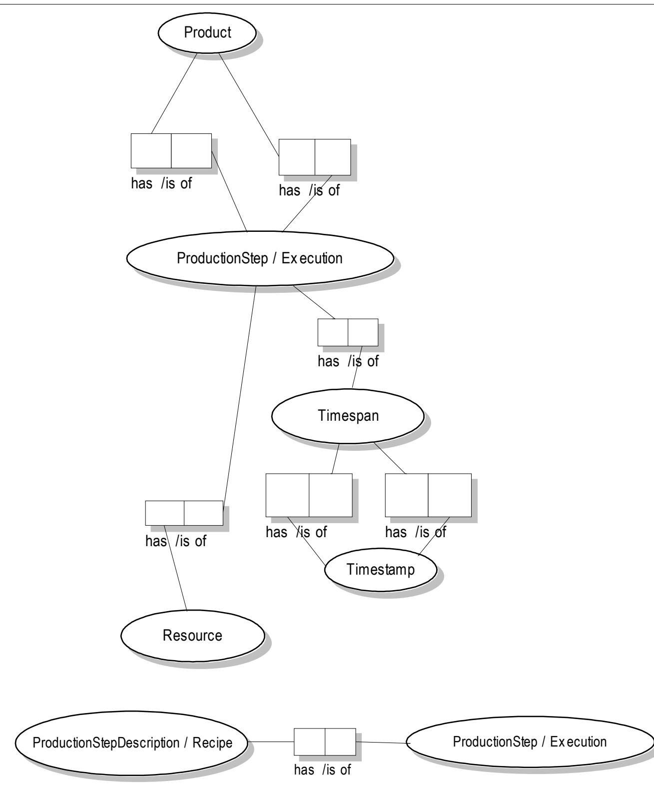
Just like product requirements are satisfied by entities in the real world, the resource requirements need to as well. It is fair enough to consider John a human to prepare, although his cooking skills have not be explicitly mentioned. In addition to that two other items that just have been mentioned are being made concrete. This leads to:

Resource (2 pans that John got for his 43rd birthday) Resource (an oven dish John bought at Walmart) Resource (5 spoons John borrowed from his neighbour) Resource (John's advanced oven) Executer (John himself)	

At first glance John himself is a resource as well, but because this definition is not always considered ethical or because one can think of cases in which a distinction between resources (which are non-autonomous) and humans (wich are autonomous) is wanted a new term executer is introduced. The executer becomes a subtype of a resource, making it a special kind of resource.

Executing the order

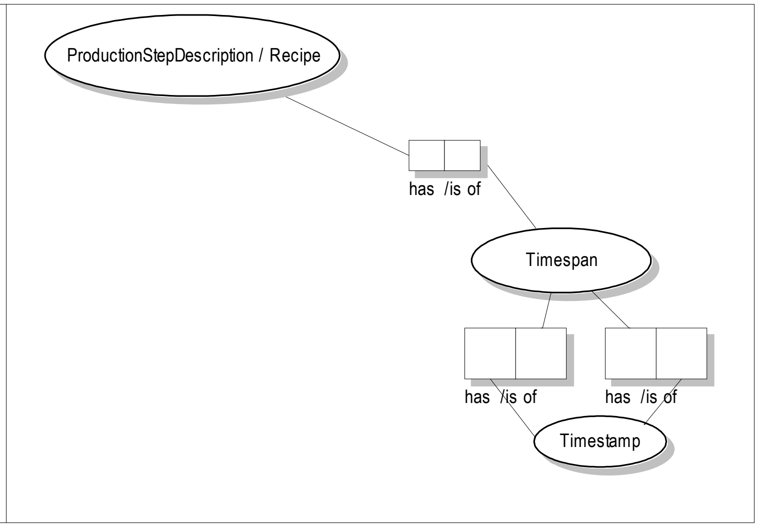
Now that everything is settled for preparing the meal, John starts doing it at a planned time. In this case this would be at time 16:45 at Valentine's day, in order to give him 15 minutes of slack with the 18:00 deadline. This results into a production step that is supposed to satisfy the previously mentioned production step description, also labeled execution (note the similarity with executer). The production step is the place where everything that is planned and thought of in the abstract world comes alive. The resources and products are linked to it and are all occupied for a certain amount of time. The time span is conceptualized here using a beginning and ending time, but other solutions are possible such as a set of the smallest time span possible in the system (which implies the assumption of time being discrete, at least in practice). In this fictional case it takes John 1 hour (as estimated) to complete the meal. The following conceptualization occurs:

ProductionStep(John spending time in the kitchen at valentine's day) Timestamp (16:45 at Valentine's day) Timestamp (17:45 at Valentine's day)	

Planning and control

The planning, even though just mentioned, needs to be formalized as well. Since a planning occurs in the abstract world and complexity needs to be reduced in this world, the planning deals with thinking in terms of production step descriptions opposed to actual production steps. Also one does not know the actual product that is going to be produced, since everything that is planning is supposed to happen in the future. The best that can be done then is mentioning the requirements this production step needs and hopefully is going to fulfill. Just like the production step above, the production step description is connected to a timespan, namely the time in the future it is meant to happen. This is pictured as follows:

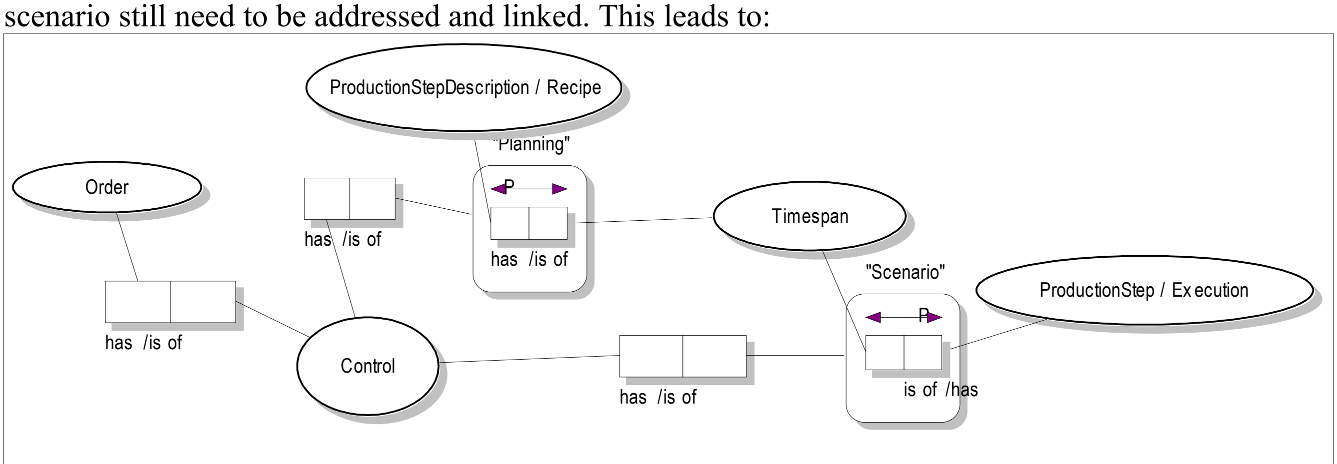
Recipe (How to make Scottish Kale oven dish)
Timestamp (16:45 at Valentine's day)
Timestamp (17:45 at Valentine's day)



Finally the control is introduced. The control is an entity that receives information about the orders that are coming in and the production steps that have been taken in the past along with their progress. The task of the control encompasses the need to decide what the planning should look like. The production steps that have been taken which are relevant for control are called a scenario and therefore control can be seen as a mathematical function with the following signature:

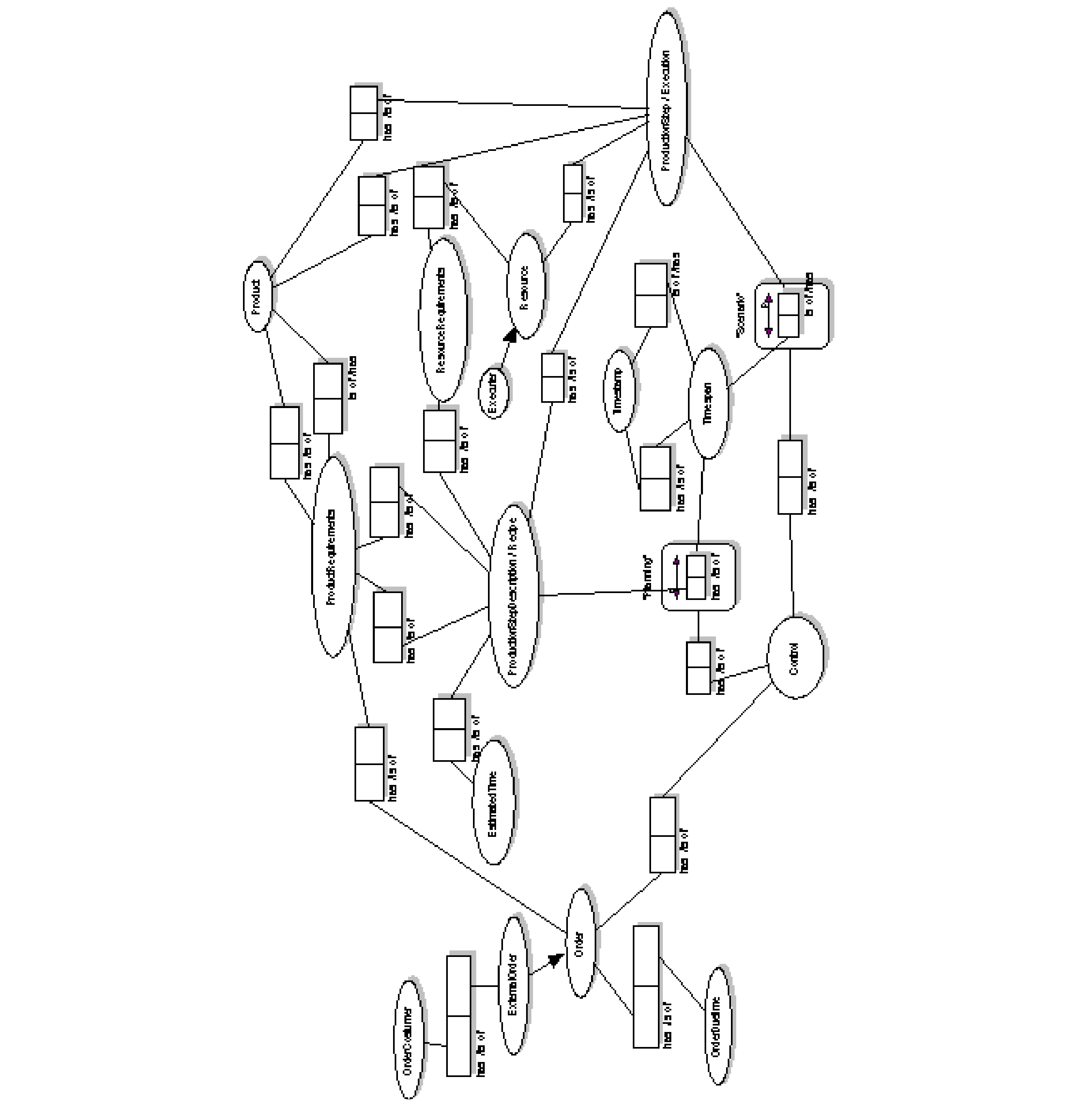
Control :: (Orders, Scenario) → Planning

The orders are already part of the iterative meta model that has been build so far, but the planning and scenario still need to be addressed and linked. This leads to:



The final meta model

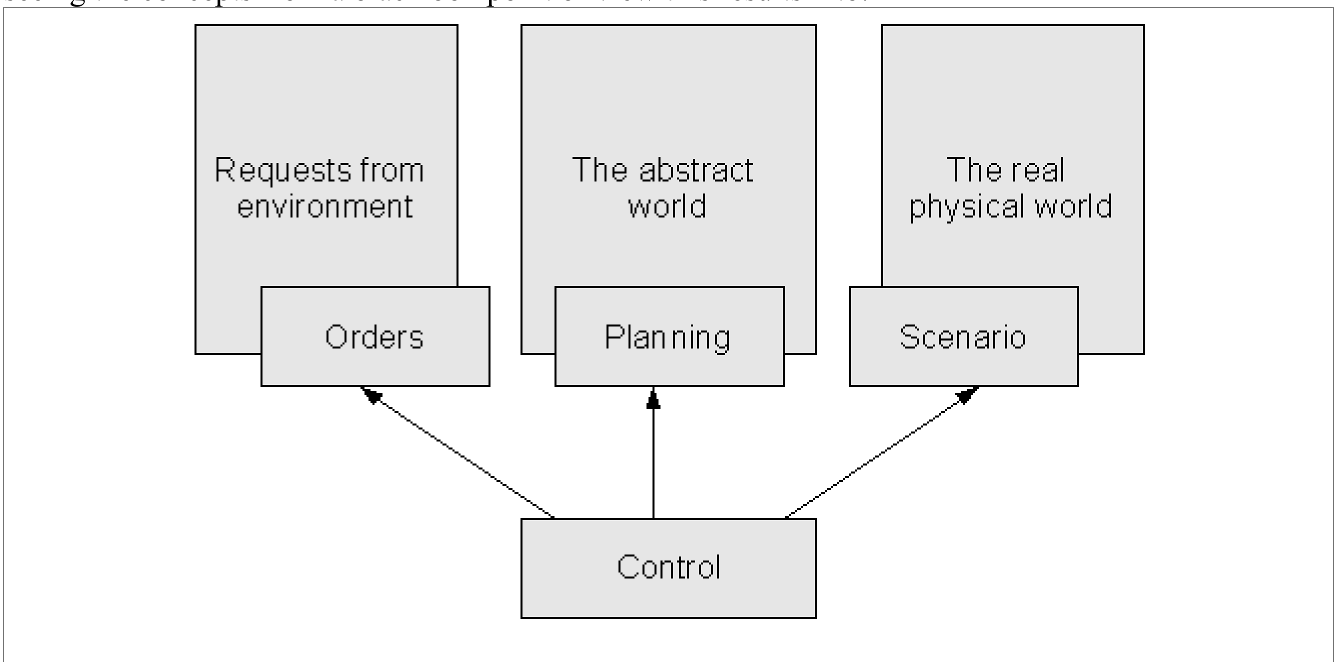
Every step taken in this paragraph has contributed to the meta model iteratively. Combining the contributions of every step, while taken into account the overlap in concepts, leads to the final model as below:



The just shown model can be downloaded for educational and scientific purposes in MS VisioModeler format (which is a program by Microsoft that can be freely used but is not supported anymore) at the following URL: http://freek.vdberg.us/download/bthesis.zip

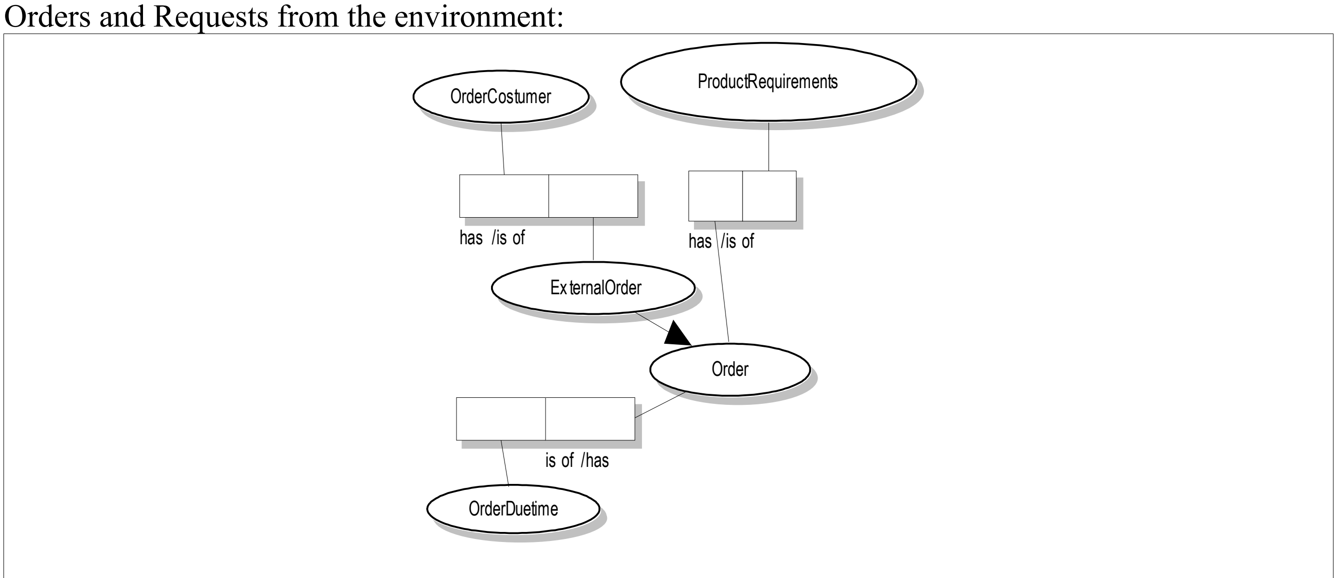
Generalized final meta model

Because the meta model contains many object types and relations between them, it might in some cases be helpful to make a more general version of the model. The main point of the model turns out to be that control is something that needs to deal with three subsystems, namely: The environment having a certain wish (orders), the system being able to perform certain actions (the abstract world) and something really going on in the system (the real physical world). Expressed in diagram form and seeing the concepts from a black box point of view this results into:

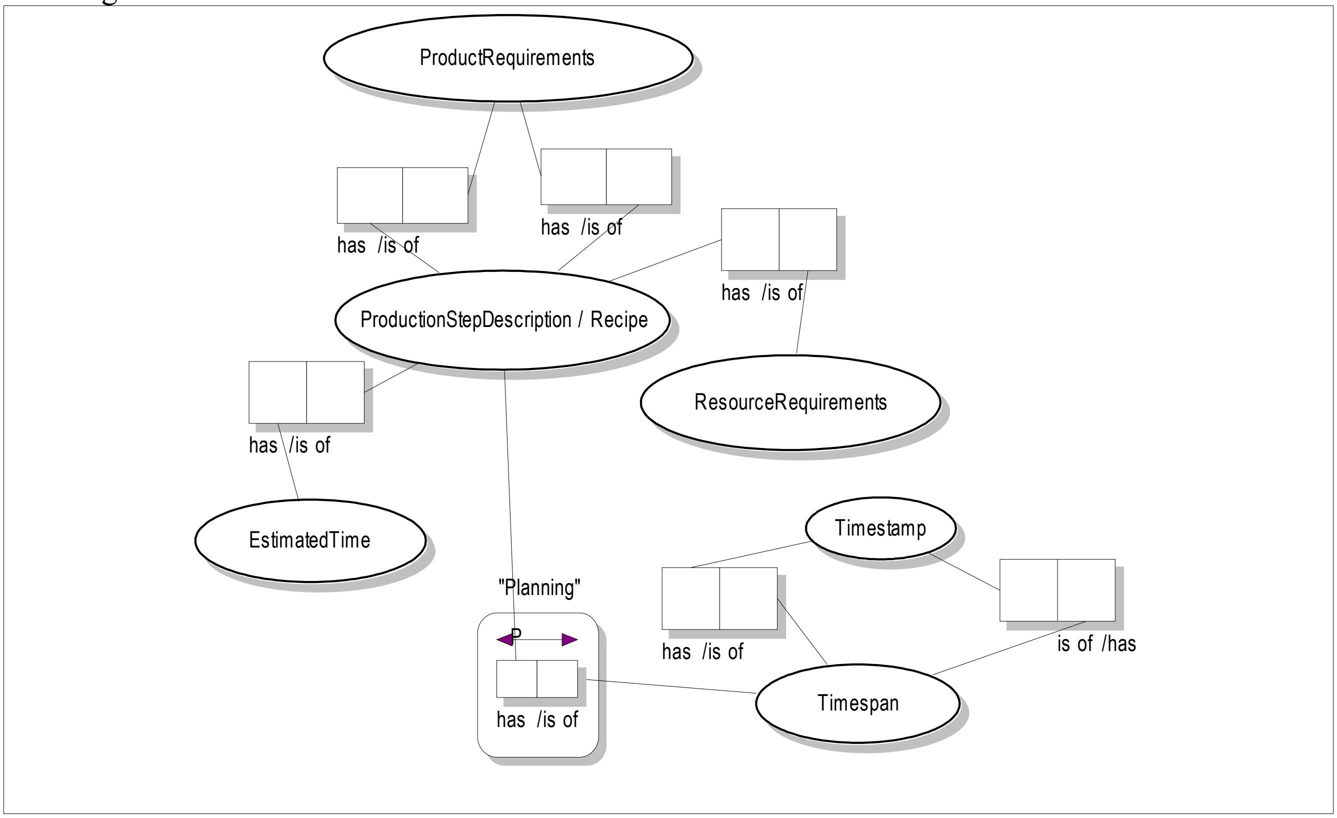


For every concept in the above model one can then address a subset of object types and fact types from the whole model which deal with the concept. This leads to the following three sub meta models.

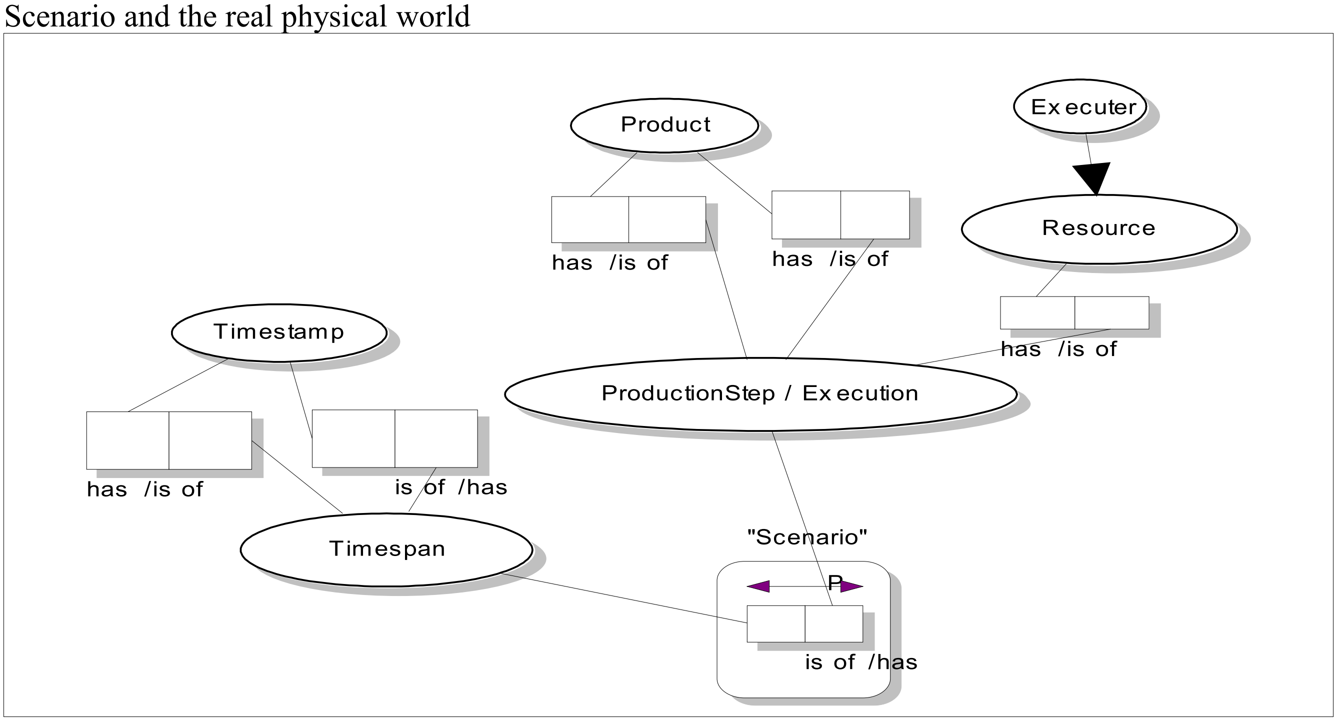
Orders and Requests from the environment:



Planning and the abstract world



Scenario and the real physical world



Note that the three sub meta models are implicitly connected as can been seen by looking at the connections of the concepts in the main meta model (eg product and productrequirements).

Design decisions

During the creation of the model in this chapter, several design decisions have been taken and although not mentioned in the chapter different alternatives to them have been considered.

Firstly the language in which the meta model has been expressed (also called the meta meta model) named ORM[6] standing for Object-Role-Modeling. There are several reasons to choose this modeling language of which three are:

- ORM models are completely formal and therefore can be completely expressed in terms of mathematical building bricks. This makes calculations and reasoning with the meta model in an unambiguous way possible.
- ORM models can be transformed to other models of which relational data base models are very useful for storing data in real system.
- ORM is made for describing objects and their relations, which is a powerfull way to connect different concepts of a taxonomy

Secondly the iterative building of the model. The bottom up approach has been chosen to illustrate clearly what every small change in practice means to the model. This approach is the opposite of a top down approach in which concrete actions are harder to link to abstract concepts.

Thirdly using a fictional but concrete example to describe the metaphor. By using the example about John, it is possible to populate the ORM model right away as actions occur. This causes the modeler to think critically about what to put in the model, because he can not just get away with naming concepts and not being able to provide a sample population.

Fourthly the lacking of constraints such as uniqueness and total role constraints in the ORM model. Although ORM provides powerful constraints to keep the population within desired limits, the introduction of those constraints will bring up philosophical issues, such as:

- Can a production step involve no resources?
- Are there products without requirements?
- Can a production step description involve no actions and can the output of a production step description be equal to the input?

It is too early to decide about what the answers to such questions should be and there might be situations in which the answers are different from what is considered normal for various kind of reasons.

Fifthly not using labels in the ORM model. Labels are a strong way to connect what is in the meta model to what is going on in reality. Normally the labels contain entities that can be stored in a computer, such as strings, booleans, integer numbers, floating point numbers and dates. Also entities that arise from combining these build bricks are optional. The values they should contain at a certain time are determined by observing the real world and translating the relevant observations to something that can be stored in a computer. During the building of the meta model, this phase has been completely omitted, because different situations can have very different requirements. Also by already committing oneself to a certain kind of label, the way one should measure things from the real world is very predetermined, while leaving it open makes the meta model much more powerful. What is been done so far when mentioning the population is using common sense terms so that for a human it should be clear what has been talked about in reality.

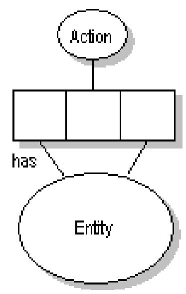
Sixthly lacking the option to apply composition/decomposition in the model in order to support what is mentioned about work flows in the current literature more. The next paragraph will mention some things about it, but the main focus of this thesis is addressing different concepts with respect to describing (and ultimately managing) a production system formally and how these concepts are related. Composition/decomposition is only a matter of how detailed a system is being observed at a certain point of time then.

Composition and decomposition

What has been missing in the model so far is the principle that entities consist of smaller entities. Some examples that match the example of John's cooking store from before are: 500 grams of Scottish kale can be a composition of 2 bags of 250 grams, 1 hour consists of 60 minutes, the final meal created consists of all the ingredients that were added to it and preparing a meal consists of all the single subtasks that are needed to get the meal on the table. The just mentioned examples show that the meta model as currently presented is missing something essential.

In order to add the needed functionality the Chinese box principle is used. The name Chinese box is used, because Chinese boxes are boxes of graduated sizes that fit in each other. The metaphor with decomposition is that one can open a bigger box and find more smaller subcomponents inside. The other way around one can decide to not care about details of a smaller box and to put different boxes in one bigger box and close the bigger box.

However, just saying that a bigger entity is just the sum of the smaller entities is not enough. For example consider the example of a bike consisting of different parts. If one needs to be in the station in 15 minutes, it will not be good enough to provide one with all the parts that make up a bike. After all it will take a person a long time to put a whole bike together. Because of this a composition consists of different parts plus an extra effort needed to put them together in some sort of way. As a result of this every entity in the model can be folded open and replaced by the following construction:



In the above picture the action is something that has or needs to be done to put the different parts together to make a newly composed entity. The entity has two connections to the fact type of which one is for indicating that an entity is part of a bigger entity and the other one for stating that a bigger entity consists of certain parts. Note that when there are different layers of abstraction an entity could be involved in both relationships. A concrete example is a dish with food on a dinner table. The dish with food is part of the whole meal (thus a dish with food is a decomposition of the meal) and a piece of meat is part of the dish with food (thus a dish with food is a composition of different parts of food).

The just mentioned is not part of the general model that has been provided in the thesis, but is meant to just give an impression what problems one can run into when applying composition and decomposition and to provide some inspiration about how to deal with it. For example, the just mentioned can not deal with extra efforts needed to decompose a physical object, while in some environments (eg. a car junk), the value of the separated parts is higher then the object as a whole. Also different entities in the model might require slightly different approaches with respect to composition and decomposition, depending on the way the model is used and in what environment.

Validity of the meta model

Now that the meta model has been built based on the given literature, it should be possible to look at the meta model and apply or recognize the aspects that are mentioned in the literature in the model. This is a way to achieve at least a certain amount of validity of the meta model.

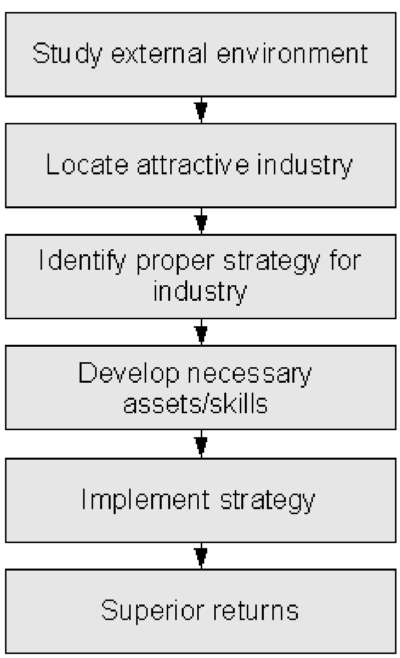
System Theory

Firstly Ashby's system theory. What Ashby mentions is that a system has to deal with disturbances being able to provide satisfying regulating actions. The disturbances that occur in the meta model are scenarios not matching the planning. The regulating action that can be taken as a respond to that is to change the planning. This is all done by control which manages the system. Ashby's theory puts emphasis on taking away complexity. The reduction of complexity occurs when a translation from the real world properties to abstract properties is made. For example when a production step is generalized to a production step description. This results into different production steps being considered all of the same kind, even though every situation is unique.

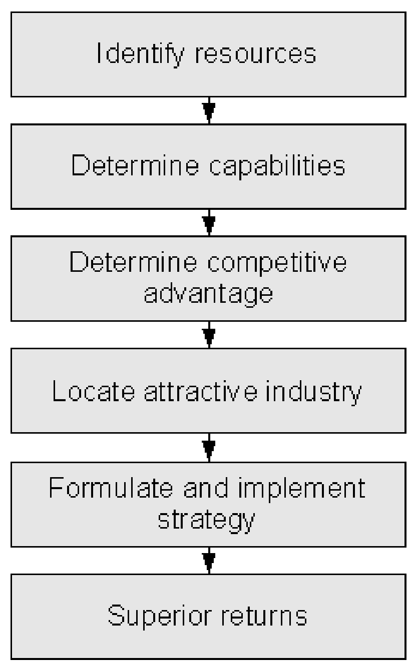
Model of superior returns

Secondly the I/O based and resource based model, which are about satisfying the environment. In the meta model a wish from the environment is expressed by an incoming order. By planning right and keeping an eye at the scenario all the time, one can calculate how as many orders can be satisfied. This includes the time in which a product is produced and at what price (the usage of the resources). Also anticipation about what orders might come in the future is something that can be taken into account. In addition to what has previously been mentioned about both models, there are plans available for how to deal with both of them. Those plans include several steps to take in process form and are as follows:

I/O based model



Resource based model



In this thesis the environment is not taken into account and therefore the steps that include the industry have been omitted from explanation here. The steps that are relevant and their translation to the meta model are:

- Identify resources: Check the population of the resource object type in the model and see what these resources can perform by looking at the corresponding links to resource requirements.
- Develop necessary assets/skills: Make sure to have or obtain resources that fulfill a certain resources requirements
- Formulate and implement strategy: Create a certain structure in your planning that can deal best with the orders that are coming in and the order you are about the expect.
- Implement strategy: Continuously compare scenario and planning and have them match the set of orders you would like to fulfill.

Verification model for embedded systems

Thirdly the verification model for embedded systems. This model connects the abstract and real world and states that if the model is right and one accepts some assumptions to be true, the result will be as one wishes. In the meta model of this thesis this is translated in the following way: When input products of a process and resources required for it are available and when the process produces the right result as described in production step description, the result of the process will be the product one wanted to make to satisfy an order. When we look back at the diagram as shown in the literature study of this thesis, we can translate the different concepts of the verification model to parts of the meta model of this thesis. This is done by listing the concepts of the verification model in the following table and mentioning the matching concept of the meta model of this thesis between brackets:

Blue print, verification model (The population of the physical world part of the model)	Proof (Linking the population of physical world part of the model to the abstract one by showing that the real world products and instruments fulfill certain requirements	**Formal properties** (Figuring out what you formally want to happen and what is possible at all. This is the control consisting of orders, recipes and planning)
Modeling (Translating the measurements from the real physical world to labels)		Formalizing (Figuring out what is happening in the real world in terms of formal properties)
Embedded system, artifact (Detecting and measuring the concepts of the meta model in the real physical world. For example: what are my resources?)	Confidence (The actual production steps happening will provide the result you have aimed for)	**Behavior, requirements** (What you wanted to happen will happen with the physical world)

Kitchen metaphor

Fourthly the use of the terms of the kitchen metaphor. Every term mentioned in the current literature has either been addressed or can be placed in the meta model of the thesis by looking at the description in natural language. Because of this the meta model is useful and sufficient enough for the following activities:

- Providing a formal definition for every kitchen metaphor term
- Figuring out how the terms of the kitchen metaphor relate to each other

To illustrate this a bit more, the concepts that have been produced while creating the meta model are listed in the table below here with a description in natural language. The description has been derived from the meta model by seeing what other concepts the specific concept relates to. This includes both concepts inside and outside the system (eg. an order is connected to something outside the system, namely a wish from the environment). This leads to the following glossary:

Term	**World**	**Formal description in natural language**
Order	Abstract	A wish to obtain a product meeting certain requirements at a certain time by a customer
Product requirements	Abstract	A requirement an actual product does or does not fulfill
Product	Real	A physical entity in the world that might be used to fulfill an order or as an input for a production step
Recipe	Abstract	A description of transforming product(s) into different product(s), aided by using instruments
Production step	Real	A transformation in the real world from product(s) to new product(s) using concrete resources
Resource	Real	An entity that has the ability to transform products when engaged in a production step, but can only do one or a limited number of tasks at the same time
Resource requirements	Abstract	A requirement an actual resource does or does not fulfill
Scenario	Real	A subset of all the production steps having something in common, such as a timespan, location and/or resource involved.
Planning	Abstract	The scenario one wishes to occur in the future
Control	Abstract or Real	The entity mediating between order, scenario and planning in order to achieve what is best for the system
Timespan	Abstract	A subset of the total life cycle of the system
Customer	Real	An entity outside the system representing a wish from the environment of the system

Work flow systems

As mentioned before in the literature chapter, the literature of three papers regarding work flow systems will be used for validating the model.

Firstly Van der Aalst's paper from 2003 in which the control-flow perspective, a data perspective and resource perspective are discussed as a view on controlling a system. These three perspectives can be rephrased in terms of the model as proposed in the thesis. This is done in the following way:

Perspective	**Explained in terms of the meta model of this thesis**
Control-flow perspective	Focus activities and their execution ordering. When activities happen in a certain order, the result should turn out right: By looking at the input and output of different processes in the model (productionstepdescription) and connecting these well, one can achieve a bigger task as long as the inputs and outputs meet the standards as described in productrequirements. The key issue here is to link the different productionstepdescription items in such a way that the productrequirements of the output match the productrequirements of the next input.
Data perspective	Layering business and process data. Every piece of work actually happening needs to have a goal and meet some requirements, which can be found in the business data/abstract world of the company: In the model of this thesis this is expressed in the distinction between the abstract and concrete world (eg. product versus productrequirements). The model within this thesis can after transforming it to a relational model be used as the meta model for storing all the needed data of the system.
Resource perspective	This perspective assigns roles and responsibilities to different people in a company and when everyone does their work well, a good result will be achieved: This is done by control making a planning that is dependent on the incoming orders. By making the planning, resources are set up for doing certain tasks at certain times. When a task fails, the machine or human can be held responsible for failure and measurements (such as fine tuning machines, firing people or instructing people) can be taken for the future.

In addition to what has just been mentioned, the paper also deals with different constructions in making a flow, such as an OR or AND operation in the flow, indicating that a certain process is followed by more processes, a random process from a subset or that in order to start a certain process more or a random process from a subset need to be finished first. The model as proposed in this thesis, does not offer a way to graphically make this picture right away, but as mentioned in data perspective row in the previous table, productrequirements can be a mediator between different processes. In other words, processes are connected by matching inputs and output types, and thus every process can be seen as a transformation from type A to type B. Plasmeijer et al. have implemented this using a functional approach in a environment called iTasks[13]. When looking at this issue functionally, productrequirements can be seen as the domain of all products that fulfill the requirements as wanted.

Secondly there is W.M.P. Van der Aalst's paper from 1998, which deals with three dimensions on work flows, namely: the resource, case and process dimension. By mapping the five terms resource, activity, task, work item and case to the meta model of this thesis, one can see that the model is complete with respect to maintaining the three dimensions. This is done in the following table with corresponding explanations:

Term	**Location in thesis model**
Resource	Resourcerequirements/Resource. In the meta model of this thesis, one can with the combination of the two just mentioned concepts determine what capabilities the system can offer to the environment, independent of a certain process or case.
Activity	Control. Dealing with several work items and making sure resources that are needed don not overlap in different activities they are scheduled for.
Task	Productstepdescription. Some process that can be performed using a recipe, without caring whether one or more cases are currently running in the system.
Work item	Link between Productstepdescription and Productionstep: The combination of something actually happening in a company and standard it is supposed to meet (purpose).
Case	Scenario and Productionstep (and everything involved in it). Concrete activities going on in a system, without knowing why they are happening or what resources they are taking up with respect to other cases.

Finally the BPMN notation as explained carefully in the paper of IBM. The event, gateway, activity and sequence flow are taken from it to discuss in this paragraph. An activity exactly matches the definition of a productionstepdescription in the meta model of this thesis. A sequence flow connects them, which can be achieved by matching the productrequirements as mentioned before. This leaves the event and gateway. An event is a trigger which causes the a business process to start in order to change something. There are two main reasons in a production system to use a trigger in a production system, namely:

- A new incoming order. This can be checked by checking for changes in populations of the meta model.
- A time trigger to have control do something every now and then (eg. Calculate a new planning every hour). In a computer this is normally done by using the computer clock.

To conclude the BPMN part, the gateway is something that needs to be dealt with. An OR can be seen as executing any process after another that can do something with the input the first one created, while an AND means that one process requires the output of several processes. Once again these connections are implicitly stored in the productrequirements of both the input and output of every process.

Application of the meta model

In order to illustrate how the meta model can be used within a company (but for any other system the same conditions hold), this chapter has been devoted to abstract real life problems that occur in systems and how they are expressed in terms of the meta model as introduced in this thesis. All the problems are related to dealing with orders, having limited resources and producing something and are as follows:

- Of all the orders coming in a certain percentage can surely be fulfilled
- The system has enough resources to satisfy the environment
- The resources of the system are used as optimally as possible
- When the number of orders fluctuate a lot over time, the system can still handle it
- The incoming orders can be fulfilled within a certain time

In order to analyze the just mentioned problems there are two requirements. Firstly queries have to been done on the information structure to obtain the needed information to derive conclusions from. This is done by doing path expressions on the ORM structure of the meta model. And secondly calculations and/or simulations need to be done on the data structure in which the former mostly focuses on evaluating the history and the latter mostly on seeing whether the system is prepared for the future. For the latter case system dynamics would be a good way to simulate what would happen in the system due to an approach in which a mix of qualitative and quantitative aspects are combined.

Fulfilling orders

Going back to the problems just mentioned the first problem is about fulfilling orders. The fulfilling of orders is normally constrained by two things in a production system, namely the resources available and the effectiveness of the production steps. The resources that are available can be read from the planning in the system. In case at some time more resources are needed, a different planning could be considered. A rule inference engine can be helpful in finding a solution that satisfies the criteria. With respect to the production steps one can assume that the production step description and the planning are appropriate enough to make safe predictions. In order to guarantee more safety some padding can always be added between events (since production step do not always lead to the results aimed for). In addition to that improving the production step descriptions can be done as well. In this case experience from actual production steps that are performed in the past can be monitored within the meta model and aid in seeing what the result of the production steps really is.

Sufficient resources

The second point has already been revealed a bit in the previous paragraph and is about enough resources being there to fulfill the orders. When it turns out that not enough orders can be fulfilled at first and when better planning does not help as well, the meta model can provide information about what extra resources are needed. Once again calculations, simulations and path expressions in the data can provide the answer here. In case of the path expression one can look at the orders that cannot be and/or have not been fulfilled and what resources they require. An investment in expanding these resources would be a serious option then, but also looking at how to use these resources more efficiently is worth elaborating in.

Utilization of resources

Continuing with the resources and the third point as mentioned before, the utilization of the resources could be considered as well. Especially when there is a cost attached to using the resources, one wants to be sure that the resources are used as optimally as possible. This includes resources being in use most of the time and perhaps even resources being used for other purposes as initially aimed for when there are no orders to fulfill (eg. an employee temporarily being placed to a different department when there is no work, even when this new department is something he is not familiar with).

Fluctuation in the number of incoming orders over time

Fourthly the fluctuation in the number of incoming orders over time, also referred to as jitter. When the number of orders in a system is constant over time, it is easy to anticipate to them and know what to expect. Unfortunately in most real systems this is normally not the case. Some examples include: more ice cream is sold when the weather is nice, the Internet is used more when businesses are open and more people do their tax declaration when the due date is coming closer. The consequence of this is that resources might be idle for a long time and then suddenly become overused. By using historical data that has been generated, one can predict what is going to happen in the future better and know how to anticipate to such situations. The data can be viewed by looking at the previously generated population of the model, since the time dimension is included in the model. The anticipation to fluctuations can be done in several ways of which a few examples are:

- temporarily have more resources by hiring them
- create virtual internal orders so that stocks are created
- higher the prices of orders charged to the customer temporarily so that the amount of orders reduces
- refusing a certain selection of orders in advance.

Order time

Finally the time in which orders can be fulfilled is discussed. This fulfillment of the orders can be expressed mathematically in many ways in which the following points are just some main examples:

- The average time between getting an order and producing what needs to be produced for it for all orders
- The median time between getting an order and producing what needs to be produced for it for all orders
- The order with he maximum time between being placed and producing what needs to be produced for it
- The maximum stock size over time

All the points just mentioned deal with two major aspects that can benefit from fine tuning:
Firstly the environment (eg. a customer) doing the order can be served better, because orders can be served closer to the time they are needed. Secondly the system itself can benefit from having to deal with smaller stocks and having to keep a smaller backlog of orders, which in some way saves need for resources. Once again the above points can be determined through calculations, looking back in the history of the system through path expressions and simulating what might happen.

Future work and improvements

This thesis has provided a framework for dealing with a production unit in a formalized way by using the kitchen metaphor. The abstract basis has been provided, but there's still many details to fill in to make it operational in real systems. In addition to that improvements to the model and making mathematical statements about it is still something that requires attention. Therefore this chapter has been divided into three paragraphs dealing with every single category of the improvements.

Operationalization in practice

First the operationalization of the meta model. The operationalization deals with determining how the meta model can really be used in practice. When the meta model was being built in this thesis it observed John performing actions in the kitchen and translated them to a population of entities in the meta model. Even though the model has now clearly and formally been defined, recognizing the concepts within a system might still be a burdensome procedure. Some examples include: What are the products a bank is selling? What can be considered a production step description in a computer system? What is considered an order in a supermarket?
Likely answering these questions will require a domain expert to engage in observing the corresponding domain. For future work it might be a good idea to figure out what is needed to provide this domain expert with a good set of tools for observing the domain and recognizing all the concepts needed. The creation of modeling tools, an example system or a set of procedures might be the way to go here.

After the right concepts have been addressed, they need to be translated to the abstract world. What has already been mentioned in the paragraph about design decisions is that this requires labels to be attached to the entities of the meta model. This is needed to be able to store in a computer what you are talking about. The labels need to address a property of an object in the real world that distinguishes the object from all other objects. For example: A certain social security number is only given to a certain person maximally once and therefore talking about a specific social security number is enough to know what person is being talked about. It is worth investigating how bar codes and future technological developments such as RFID can play a role in this.
In addition to this one can investigate whether and to what extend some entities that were or are still in the real world can be translated to completely abstract formal entities. An example here is the recipe that John used to prepare the meal which has been found on the Internet and thus has not even been part of the physical world from the start.

Once the connection between the abstract and real world has been established it is needed to keep the connection up to date. As mentioned in the Verification model by Mader et al this is done by measurements in the real world. What can be investigated is what methods there are to keep the labels (the population of the abstract world) up to date with the real physical world and vice versa in terms of sensors and actuators.

What has been mentioned in the literature study with respect to the I/O model and the resource based model and also in the framework of Ashby's model, is that looking at the environment and figuring out what is wanted by the environment is important for a system or company to survive. In this thesis these aspects have purposely not been taken into account. For future study one can figure out how wishes from the environment can be translated to the meta model of this thesis. The main focus here should then be to predict what orders one can expect to have depending on the quality of the resources of the system (read: what one has to offer to the environment). Also how to upgrade the resources to satisfy the environment is something to take into account.

As as final part of this paragraph it might be worth having a look at system dynamics. As J.Sterman states in his paper[7], system dynamics is a mix of quantitative and qualitative analysis and can be used to run simulations with numbers to guide policy or to observe a certain phenomenon, such as an exponentially growing backlog of orders over time. Since this thesis already mostly has dealt with the qualitative part of a meta model, a system dynamics model can be derived from it relatively easy. After this simulations can be ran to find out properties about the system.

Improving the meta model

Secondly the part about improving the meta model. This includes adding new concepts to the model as well as improving the current concepts of the meta model and is therefore mostly a qualitative activity. The addition of concepts could happen through observing new phenomenons in a production system that are worth addressing, such as the costs of using resources, a production step causing waste or the composition/decomposition as mentioned before. But this could also involve labeling concepts differently in the currently already available meta model. Labeling concepts differently includes adding synonyms for different domains (eg naming a production step a task) and labeling connecting fact types with a name, an activity referred to as objectification in ORM. An example here would be to label the fact type that connects product with product requirements as product testing, since making a connection between the two would involve checking whether a product in the real physical world has some sort of desired properties to be considered a certain product in the abstract world (product requirements).

The improving of concepts can be done to have domain experts from different domains, which of course are still production system related, look at them critically and try to have some sort of consensus between them about what the inter subjective truth among them is. Also applying other related theories to the model and seeing what is missing or wrongly defined can be of added value.

Finally one can observe that the meta model that has been introduced so far only includes deterministic outcomes. It might be worth paying some attention to adding probabilistic outcomes to the meta model. For example, when some input products are processed using some recipe the formally defined output product is not always the result of this transaction. Not even when the input products and the instruments involved in the whole process meet all the requirements. Of course control can deal with this after the mistake has been detected by changing the planning, but that would be rather late and chaotic, especially if this is often the case. Instead of that it would be better for control to have information about the risks some steps involve (due to external effects that have not been modeled) and then select the best way to go while keeping all possible risks in mind.

Mathematics

Finally the mathematical aspect. Although not mentioned and showed too often in this thesis, it is a fact that ORM offers many ways to calculate with models. The first thing that can be investigated is that if it is possible to express the concepts as introduced in terms of the kitchen metaphor can be conceptualized in a mathematical way. This can offer two major benefits:

- The number of terms can be reduced, because many terms can be explained as a composition of others terms
- A signature for different functions can be derived that fulfill some sort of action in terms of the kitchen metaphor. For example: a production step description can be a function that takes a set of products and a set of resources as input and returns a (normally different) product.

Predicate logic seems the right way to go here, since this kind of logic has contributed a lot to how ORM has evaluated until now as made clearly by Terry Halpin in several papers and books[6].

Two paragraphs ago the the operationalization was mentioned and it turned out that a population needs to be stored in the abstract world to reflect the real physical world. Through the meta model of this thesis having ORM as a meta meta model, the shape of the population is already defined. The way it is physically stored has not been defined yet. Two options that are commonly used are XML and relational databases. In both cases there are different solutions for shaping the population, depending on what access (read, write, delete) is needed. P. van Bommel deals in his lecture notes called 'Data models and transformations'[8] with these issues. It is possible to investigate if there are common demands in different production environments and if so what the data structures should look like.

While working with the ORM structure as proposed in the thesis, it should be possible to detect certain path expressions that are commonly used and are perhaps of some value. These path expressions could be gathered and labeled. In addition to this new rules the system needs to comply with can be addressed, also known as complex constraints in ORM. An example here is: When a production step description explains how to create a product with certain requirements, then an initialization of the production step description (read: a production step) should result into a product fulfilling these needs. Some constraints might be there for all production systems and some needs are domain specific or even system specific. For the more specific rules a business rules approach can be considered, making it possible too loosely couple the policy from the system as mentioned by A Charfi in the paper called 'Hybrid web service composition: business processes meet business rules'[9]. Also these business rules can be formally expressed in terms of constraints on the population of the model.

Finally there is still mathematical work to do with respect to work flows. As seen before different processes can be connected through having the output of one or more processes matching the input of a process later in time. When all the information is available about the possible processes (in the meta model in Productionstepdescription) and at the same time an order is known, one can try to figure out different ways and internal orders in the process diagram to fulfill this order. In order words, using advanced algorithms one can use all the knowledge that is available within the population of the meta model of the system to have work flows created automatically. The approach that seems logical to use here, is to see the final product one likes to have as a goal state and a bunch of initial input products (normally the current stock) as a starting state. Transitions would then occur by running a process. A backtracking algorithm would then be able to run over the whole search space and find solutions.

Conclusion

This thesis aimed to formalize concepts that are currently used in a production system environment and figure out how they are related to each other.

By including a literature study, this thesis has shown that many frameworks and views that are currently available in the literature about production systems contain components that can be generalized. After this the overlap of the different theories has been taken.

While keeping in mind the results of the literature study, formalization using the concepts of the kitchen metaphor, has lead to a meta model in the ORM language in which all relevant object types and how they are related (fact types) have been expressed in a formal way. This meta model has been created iteratively using a fictional story in the kitchen, in order to see by small steps what is going on and making what is happening accessible to a big group of people.

After the creation of the meta model a generalization has been made, in order to see what is happening the meta model at one glance. It turns out that a production system is controlled by some entity called control and that the task of control is to make a planning based on two things:

- Orders: What orders are coming in?
- Scenario: What has been happening in the system so far?

Applying the different theories on the model has shown that the aspects that are relevant in the corresponding theory can be expressed in terms of the model, thus showing validity of the model.
The model has shown to be compatible with Ashby's theory by being able to address the essential variables, regulating actions and disturbances. Also the meta model of this thesis has shown in what way the complexity has been reduced, which is something Ashby aims to do.
For the I/O based and resource based model the meta model of this thesis has been capable of being able to translate several steps that need to be taken in the I/O and resource based model to formal constraints and actions that need to be followed.
The different concepts of Mader et al's paper about verification of embedded systems are visible in the model and a direct mapping between both models can be made.
The concepts that are part of the initial glossary of the kitchen metaphor have been addressed, which make this thesis useful for providing a formal glossary with which different parties can reason and communicate. To illustrate this more, the kitchen metaphor concepts as revealed in the meta model, have been translated to natural language again.
To link everything that has been done here to the current work flow literature, three influential papers have been selected and many concepts that have been mentioned in those papers have been successfully mapped to the model as proposed in this thesis.

Even though a general basis has been set for a formalized view on production systems, there is still future work remaining in this area of field. This future work includes:

- Improving the meta model qualitatively
- Making it possible to apply the meta model in practical
- mathematical reasoning with the meta model.

For every step several suggestions have been given.

References

[1] A. Mader, H. Wupper, M. Boon, The Construction of Verification Models for Embedded Systems, eprints.eemcs.utwente.nl, 2007
[2] W.R. Ashby, Principles of the self organizing system, Pergamom, 1962.
[3] J. Tirole, The theory of the industrial organization, MIT press ISBN 0262200716, 1988
[4] M.A. Peteraf, The Cornerstones of Competitive Advantage: A Resource-Based View, Strategic management journal, 1993
[5] Digital learning environment Radboud University Nijmegen, The Netherlands, http://lab.cs.ru.nl
[6] T.Halpin, ORM2, Springer Berlin / Heidelberg ISBN 978-3-540-29739-0, 2005
[7] J.Sterman, Learning in and about complex systems, System Dynamics Review, 1994
[8] P.van Bommel, Data models and transformations, Lecture notes Radboud University Nijmegen, 2009
[9] A Charfi, M Mezini, Hybrid web service composition: business processes meet business rules, ACM New York, NY, USA, 2004
[10] W.M.P. Van der Aalst, A.H.M. Ter Hofstede, B. Kiepuszewski, A.P.Barros, Workflow patterns, Distributed and Parallel Databases, 14, 5–51, 2003
[11] W.M.P. Van der Aalst, The Application of Petri Nets to Workflow Management, Journal of Circuits Systems and Computers, 1998
[12] Stephen A. White, Introduction to BPMN,IBM Cooperation, 2004
[13] R Plasmeijer, P Achten, P Koopman, iTasks: executable specifications of interactive work flow systems for the web, Proceedings of the 12th ACM SIGPLAN international conference on Functional programming, 2007

Appendix A: Screen shot website: The recipe

In order to describe the kitchen metaphor carefully, a real recipe for food from the Internet has been used which can be found at the URL:
http://www.receptenweb.nl/SiteNL/servlet/ctl/jsp/RecipeSearch/Recipe.jsp?index=1&word=boerenkool&scope=4&isSMSSearch=false

A screenshot of the website at taken on :

boerenkool ovenschotel

Bron: Liesbeth

Heerlijk smakelijke, niet zure, maar kindvriendelijke boerenkool ovenschotel.

Ingrediënten

500 gr. boerenkool
1000 gr. aardappelpuree
1 stuk rookworst, magere
120 gr. cr?me fra?che
3 el. bieslook
2 tl. bouillonpoeder

Waardering

Lees hier de reacties van andere bezoekers en geef je eigen reactie.

Benodigdheden

Twee pannen, ovenschotel en een paar lepels.

Bereidingswijze

Je kan de rookworst heel goed vervangen voor wat je zelf lekker vind, zoals andere worsten of gehakt ofzo.

Verwarm de oven voor op 180 C.

Maak de aardappel puree. (Dit kan vers maar ook met een pakje.) Roer hier de cr?me fraiche doorheen en de bieslook. Laat de boerenkool slinken in een pan, eventueel met wat boter of olie(niet in ingredienten opgenomen) en voeg de bouilonpoeder toe. Snijd de rookworst in plakjes en bedek hiermee een ovenschaal. Doe daar op de boerenkool en daar bovenop de aardappelpuree.

Zet de ovenschaal in het midden van de oven. Na een klein half uur heb je een heerlijke maaltijd. Eet smakelijk!

Appendix B: Translation to English of the recipe

The corresponding translation from Dutch to English of the recipe that is shown in appendix A is as follows:

Scottish Kale oven dish
Source: Liesbeth
Nice tasty, not sour, but child friendly Scottish Kale oven dish

Ingredients
500 grams Scottish Kale,
1000 grams mashed patatoes,
1 peace of saucage,
120 grams crème fraiche, 3
tablespoons of chive,
2 teaspoons of bouillon powder.

Requirements
2 pans,
1 oven dish,
a pair of spoons

Way to prepare
The sausage can be replaced with something you like, such as different sausages or meatloaf.

Pre warm the oven at 180 degrees Celcius
Prepare the mashed potatoes. The can be done freshly but also using a package.
Mix the crème fraiche and chive in it.
Let the Scottish Kale shrink in the pan. Possibly by using some oil or butter (not added to ingredients) and add some bouillon powder.
Cut the sausage into pieces and cover the oven dish with them.
Cover that with the Scottish Kale and finally put the mashed potatoes on top of that.

Put the oven dish in the middle of the oven. After half an hour you should have a taste meal.
Enjoy your dinner!