Heterogeneous Goods in Transportation Systems
A study on the uses of an object-oriented approach

PER OLOF ARNÄS

Department of Technology Management and Economics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2007
Heterogeneous Goods in Transportation Systems
– A study on the uses of an object-oriented approach
ISBN 978-91-7291-944-0

© PER OLOF ARNÄS, 2007

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 2625
ISSN 0346-718X

This electronic edition contains minor revisions to the printed version, mainly regarding spelling and typographic errors. The printed version has been supplemented with an errata-list containing these changes.

Department of Technology Management and Economics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31 – 772 1000

Printed by Chalmers Reproservice
Göteborg, Sweden 2007
Heterogeneous Goods in Transportation Systems
– A study on the uses of an object-oriented approach
Per Olof Arnäs
Department of Technology Management and Economics
Chalmers University of Technology

ABSTRACT
This thesis presents research regarding the impact differences in goods type have on the control of a transportation process. It also explores how the object-oriented family of methods are used and can be used in analysing and designing transportation systems. The aim of the research is to develop knowledge on how to control a transportation system, capable of accommodating the requirements of heterogeneous goods and to identify key factors for the design of such a system by using object-oriented methods.

The theoretical framework that is used is based upon three approaches. The first approach is that of heterogeneous goods, of how differences in goods type may cause cost increases in transportation systems. The second approach is that of systems theory in general and of cybernetics in particular. The third approach is that of object-orientation. Object-oriented methods are prominent in computer programming mainly because of their aptness to efficiently partition even the most complex of computer programs into manageable, interactive, and scalable code repositories – objects.

The main task when controlling a transportation system is to handle complexity. Transportation systems are inherently complex, and this complexity is divided into three types: descriptive, computational, and uncertainty-based. The control of a transportation system is divided into three control scopes, each corresponding strongly to one of the three complexity types.

In this thesis, it is found that heterogeneous goods increase complexity in transportation systems, and that this complexity mainly manifests itself as uncertainty-based. When designing a transportation system, it has been shown that object-orientation can be used to reduce complexity and to embed mechanisms that contribute to further complexity reduction. Eight case studies were performed and the cases were modelled and analysed using object-orientation. This resulted in a class library that is a synthesis of the object models of all the cases and consists of diagrams that evolved during the modelling process, as is consistent with the object-oriented techniques. It is concluded that object-orientation as a modelling method is well suited for analysis and design of transportation systems.

Keywords: transportation, heterogeneous goods, systems science, object-orientation, complexity
For my family

“The best way to become acquainted with a subject is to write a book about it.”

Benjamin Disraeli
British politician (1804 - 1881)
PREFACE

The work behind this thesis has been a part of my life for the last eleven years. To adequately describe how it feels to actually hold the finished results in my hand is nearly impossible. On one hand, the work can be objectively described as the sum of many sleepless nights in front of a computer, debates and discussions with colleagues and supervisors, interviews, seminars, courses, well-written and not-so-well-written literature that I have read and hours upon hours of pondering. On the other hand, the work can be more subjectively described as the hardest and most interesting thing I have ever done. In the last few years, friends and relatives have frequently asked me two questions. The first: “When will you finish your PhD?” can now finally be answered. The second: “Would you do it again, knowing what you know now?” is trickier. Ask me in six months.

The cover of this thesis holds only my name. Nothing can be farther from the truth. I have several people to thank for their contribution. Every one of you is important, and none of you could have been replaced.

Professor Kenneth Lumsden, my supervisor, thank you for your endless patience with my somewhat slow approach to research and your uncanny ability to always be able to pinpoint the weakest part of my arguments and my texts. You opened my eyes to the world of logistics during my student years and have for the last eleven years continued to challenge my mind through countless discussions and debates. It has been a privilege to work within your research team and to be part of work that is often more than a decade before its time.

I strongly believe that research is a creative process, and that intuition and inspiration are crucial to the results. Thank you Dr. Lars Hultén, my assistant supervisor, for being directly responsible for several of my “intuitive leaps”, and for always finding the right question to ask to keep me up all night writing. Lars, you are one of the few people I know who actually understands and, more importantly, extrapolates from my rather opaque and sometimes vague ideas on how the world is put together. Communication is easy when you speak the same language – when you have a common interface ;-)

Associate Professor Johan Woxenius, also assistant supervisor, is one of the world’s foremost experts on transportation systems and I have been honoured to have him and his expertise at my disposal during my thesis work. Apart from being a personal friend, Johan is also a master research craftsman and has helped me through some of the hardest parts of writing my thesis. Thank you!

I would like to thank Professor Lars Sjöstedt, who has supported me during several stages of my eleven-year journey, not least as a “devil’s advocate” during the final phase of this thesis work. With an agile, crystal-clear intellect and a genuine interest in my sometimes not-so-well-articulated theories and musings, Lars has participated in several long discussions which have resulted in major contributions to the thesis.

Associate Professor Per-Olof Brehmer of Linköping University, thank you for your thorough and insightful analysis of my manuscript – not once but twice! Your feedback was instrumental in the final rewrite which resulted in the version that you are now reading.

Phil. lic. Göran Stjernman accompanied me on several of the case studies. During the first years of my doctoral studies Göran and I had several productive and lengthy brainstorming sessions. Thank you, Göran! I still have my notes from these sessions and can, in retrospect,
only congratulate us for being so right, so early (and rebuke us for not recognising it at the
time…).

Thank you, lic eng Per Medbo, who has assisted me in the calculations for the visualisation
example in Chapter 6.

I would also like to thank Professor Håkan Torstensson of the University College of Borås,
who contributed as an assistant supervisor on my licentiate thesis in 1999. The work from that
thesis is now integrated within this document.

Ola Hultkrantz, Jonas Waidringer, Gunnar Stefansson, and Pehr-Ola Persson, you have all
contributed to my work in your own ways. Ola and Pehr-Ola: I will be waiting with Jonas and
Gunnar in PhD-country…

Professor Inge Svedung of the University of Karlstad participated in one of the case studies.
Thank you for a productive cooperation.

To my former colleagues at the Department of Logistics and Transportation; thank you for
good discussions, good feedback, good company, good coffee, and bad jokes (you know
who…).

My mother, Lille-Mor Arnäs of LM Lingua, has corrected my English. I thought that she
would let me off easy because of the family ties, but no such luck… As a result of this, any
errors left in the text are due to my own last-minute changes.

I would also like to thank the companies and individuals that allowed themselves to be
interviewed and studied. Reality is a fascinating place. Perhaps I will spend some time there,
now that I’m done with this…

This thesis has been supported financially by the Swedish Rescue Services Administration,
the Swedish Maritime Administration, Stiftelsen Sveriges Sjömanshus, Transforsk, and TRB
Sverige AB. I owe all of you my sincere gratitude.

Last, but in no way least, my family: Lotta, Jonathan, and Marcus. You have all contributed
more than you know! I love you!

Åsa, 19th of April, 2007
TABLE OF CONTENTS

1. Introduction .................................................................................................................. 1
   1.1 Forwarding problems instead of goods – what happens when past promises are ignored and priorities change? ........................................................................................................ 1
   1.2 Aim and scope of research .......................................................................................... 3
   1.3 Structure of thesis ..................................................................................................... 15
2. Methodology .................................................................................................................. 19
   2.1 Knowledge creation ................................................................................................. 19
   2.2 Tools ......................................................................................................................... 26
   2.3 Research design ....................................................................................................... 29
3. Frame of reference ........................................................................................................ 35
   3.1 Transportation ......................................................................................................... 35
   3.2 Systems science ....................................................................................................... 51
   3.3 To describe and design a system .............................................................................. 55
   3.4 Object-orientation ................................................................................................. 63
   3.5 Object-orientation applied to transportation and logistics .................................... 72
4. Building a framework .................................................................................................... 77
   4.1 A regulator model of a Transport Control System ................................................. 77
   4.2 The object as an information processor .................................................................. 81
   4.3 Encapsulation, constraints, and interfaces ............................................................ 83
   4.4 Controlling the transportation process ................................................................... 86
5. Case studies .................................................................................................................. 89
   5.1 Case 1 – Transport between two road network terminals ...................................... 90
   5.2 Case 2 – Container exported to the Far East by road and sea ................................ 96
   5.3 Case 3 – Chemicals in a tank container transported by RoRo ferry, road and rail 101
   5.4 Case 4 – Road and RoRo ferry transport of semi-trailer ....................................... 106
   5.5 Case 5 – Road, rail and rail ferry transport of semi-trailer ................................... 111
   5.6 Case 6 – Internal handling in warehouse ............................................................... 116
   5.7 Case 7 – Transportation and handling of computers in a road transport terminal 122
   5.8 Case 8 - Components for automotive assembly ................................................... 126
   5.9 Summary of the case studies ................................................................................. 130
6. Analysis ......................................................................................................................... 133
   6.1 Step one: Building object models ......................................................................... 133
   6.2 Step two: Cross-case comparisons ....................................................................... 138
   6.3 Interfaces ................................................................................................................. 147
6.4 Visualisation during design – a calculation example .............................................151
6.5 Testing the research ..........................................................................................162
7. Conclusions .......................................................................................................167
   7.1 Designing interfaces .......................................................................................167
   7.2 The uses of object-oriented modelling in transportation .........................171
   7.3 A short summary of the findings ....................................................................174
   7.4 Future research ..............................................................................................174
8. References .........................................................................................................177
Appendix 1 – List of terms .....................................................................................187
Appendix 2 – Object-orientation – the basics .....................................................191
Appendix 3 – UML – Unified Modeling Language .............................................197
Appendix 4 – Reviewed papers ............................................................................203
Appendix 5 – Object models based on the cases ..............................................205
Appendix 6 – Specifics of calculation model ......................................................259
1. INTRODUCTION

This chapter addresses the aim and scope of the research as well as formulation of research questions. First, the basic problem is specified. Within this problem lie the various system operations that are performed to meet requirements posed by a consignment.¹ This leads to, with a focus on the forwarder’s problems, the formulation of three research questions. These three questions are then referred to from different angles throughout the thesis. First, a short illustrative case of how a customer can be affected by problems that should have been contained within the transportation system:

1.1 Forwarding problems instead of goods – what happens when past promises are ignored and priorities change?

A Swedish importer of health products had launched a massive campaign promoting a certain substance. In the campaign deal, the vendors received display material as well as large discounts on the product. The product, which was manufactured in southern Germany, was very sensitive to cold temperatures. When the product was to be shipped to Sweden, Germany had an extremely cold winter resulting in acute shortage of heated semi-trailers.

The immediate response from the forwarder was to cancel the transport referring to *force majeure*. This was not accepted by the importer, who referred to the contract that guaranteed a climate-controlled transport since that was required for the northern part of the transport chain. The problem was that the forwarder did not book a heated semi-trailer well in advance since there is usually no shortage. The forwarder then wanted to postpone the consignment waiting for better weather conditions. Since the campaign was under way, the importer deemed it critical that the product was delivered on time and refused also this solution. The importer had already pre-booked services of a Swedish forwarder for the national distribution to the customers. The importer had the opportunity to transport a small amount of the products himself using his own mini van, which was in the area. That did not solve the overall problem, however. The decision was then taken to send the consignment by air freight at a cost far surpassing the budgeted freight costs. Once in Sweden the product was distributed by the Swedish forwarder according to the original plans.

1.1.1 The basic problem

The predicaments that the importer above was subject to are not uncommon. Large, standardised transportation systems are very sensitive to deviations from the “normal course of events”. Customers are frequently affected by disturbances that should not be visible outside the transportation system but problems of this type are not always contained within the system.

This thesis focuses upon the problems of a *Transport Control System*² (TCS). A forwarding company, for instance, operates a large TCS, facilitating transports across large networks. The networks are expanding and forwarding companies are merging throughout the globe, thus increasing the scopes and sizes of the networks. A trend in Europe to meet the new demands is the evolution of so-called Mega Carriers such as Schenker, Küehne & Nagel and DHL. These actors pride in multi-national, multi-continent networks of tremendous size (Woxenius et al., 2001; Lambert et al., 1998 p. 243).

---

¹ For definition, see Appendix 1 – List of terms
² For definition, see Appendix 1 – List of terms
Woxenius et al. (2001) presented a categorisation of measures available for a TCS when faced with a problem regarding heterogeneous goods.\(^3\) When a customer orders a transport that may require a little extra effort from the TCS, one or more of the following occurs (Woxenius et al., 2001):

- The price goes up. The TCS charges extra for the special effort needed, even if there are no distinguishable extra costs.

- The TCS denies the customer. Active denial is not common in the transportation business, but an example of a passive denial can be found just by looking at a letter box. If your consignment does not meet the specified requirements (i.e. physical dimensions) it will not fit into the slot, and will therefore be denied transport.

- The TCS returns the problem to the customer. If a letter does not fit the letter box, the customer will have to visit a post office in order to send it. In the transportation business, the forwarder may force a customer to conform to standards, such as the use of a pallet.

- The TCS does nothing, or too little, to accommodate the demands for extra effort. This strategy may be effective most of the time, but will backfire once it fails. A dissatisfied customer will not return.

In many cases, the customer is the one having to conform to the standards set by the TCS. Because of this, there is of course room for niche operators willing to adapt to whatever the customer wants but these are by necessity smaller as well as more expensive.

### 1.1.2 Built-in conflict

A forwarding company can be divided into two parts. The first part is marketing. This part handles the customer relations and manages the financial flows. The second part is logistics and transportation. This part manages the physical operation of transporting and handling goods.

Nominally, the marketing part wants to accept the goods, even though it is heterogeneous. The transportation part, being a production system, probably lacks the customer-oriented views of marketing and does not want to handle heterogeneous goods as they tend to increase cost (Lambert et al., 1998). These types of conflict between the more customer-oriented marketing system and the cost-conscious production system are not uncommon (Keller et al., 2006; Ellinger et al., 2006; Murphy and Poist, 1996; St. John, 1991; Voorhees et al., 1988).

When faced with heterogeneous goods, the manager of the transportation system can choose between two different approaches:

1. Charge for the extra cost the goods generate for being heterogeneous

2. Change the system rules so that the goods type is included in the homogeneous definition

Regardless of what the choice is, the transportation is affected by the heterogeneous goods. In approach 1 the effect is short-term. If there are more occurrences of the same type of heterogeneous goods, however, each short-term effect will be added to the previous thus increasing the total cost. Approach 2 probably has a higher initial cost since the system may

---

\(^3\) For definition, see Appendix 1 – List of terms
need to be redesigned. Once this redesign is finished, the heterogeneous goods are regarded as homogeneous.

1.2 Aim and scope of research

This section introduces the overall aim and scope of the research. The components – the Consignment, the Transport Control System and their mutual interface – are defined as well as the basic framework for describing their properties and relationships. Complexity is brought forward as a means of identifying the effort needed to handle the uncertainties that are produced by heterogeneous goods. The section ends with the formulation of three research questions.

1.2.1 Three approaches

The research presented in this thesis is based upon three approaches.

The first approach is that of heterogeneous goods, of how differences in goods type may cause cost increases in transportation systems. Previous work in this area has been done mainly in Arnäs (1999a). The basic notion of this area is that a transportation system is designed to handle a few types of goods, and when it is exposed to goods that differ from the nominal types, cost increases.

The second approach is that of systems theory in general and of cybernetics in particular. There are countless ways to organise knowledge and to explain and/or understand phenomena. Systems science is based upon a central concept that the whole is different from the sum of its parts, thereby negating reductionist methods as a means to understand and/or explain how a system works. Cybernetics is a specific area in systems science where the concept of controlling a system is central. Cybernetic systems are regarded as input-output centric Black Boxes. These boxes, or subsystems, can be controlled by a regulator that, by monitoring the output from the regulated subsystem, manipulates the input so as to steer the subsystem to some predefined state.

The third approach is that of object-orientation. Object-oriented methods are prominent in computer programming mainly because of their aptness to efficiently partition even the most complex of computer programs into manageable, interactive, and scalable code repositories – objects. The object-oriented framework is more than a programming tool, however. It has evolved into a general systems modelling language that closely resembles the ideas found in for instance cybernetics. The applications of object-orientation in transportation and logistics research are not very common. Because of this, a part of this thesis is dedicated to the exploration of how this area has been used within transportation and logistics research in the past.

The object of study is the control of transportation systems, capable of accommodating the requirements of heterogeneous goods. The theoretical framework, which is used to study this object, is taken from systems science in general and from cybernetics in particular. By applying the methods of object-orientation to the framework of cybernetics, the object of study is analysed.
The aim of the research is to develop knowledge about how to control a transportation system capable of accommodating the requirements of heterogeneous goods and to identify key factors for the design of such a system by using object-oriented methods.

1.2.2 Transportation, state transitions and trajectories

In systems science, a succession of state transitions within a system is called transformation (Beer, 1959 p. 40), or in some cases, trajectory (Ashby, 1956, p. 217; Rosen, 1972, p. 610). From a systems perspective, transportation can be seen as a transformation process, where products are transformed from one state into another through a succession of intermediary states. In this thesis, the term trajectory will be used to denote a succession of states in a transportation system.

The transportation process is defined by the initial state, the state space, the unit load device (the goods), the handling equipment (the resources) and the trajectory (Hultén, 1997). It is implied that the goal state increases the value of the goods, but it is not a prerequisite for the model. In the model, the system transforms goods from an initial state into a goal state through a state space using resources.

The state is in the model described as a vector, \( \vec{S} \). The state space consists of all distinguishable states that the goods can assume, where the state \( \vec{S}_0 \) constitutes the initial state.
and $\tilde{S}_1$, the goal state. $\tilde{R}$ is a vector of the available resources where each resource, $r$, can be used to manipulate whole or parts of the state-vector, $\tilde{S}$. The trajectory is a function of the input ($\tilde{S}_0$), the output ($\tilde{S}_1$) and the available resources ($\tilde{R}$).

### 1.2.3 The Transport Control System, TCS

This thesis takes the perspective of a Transport Control System, TCS. From cybernetics comes the concept of a regulator that controls a system through its interfaces. The classical model of a regulated system with a feedback loop is shown in Figure 3 below. The regulated system has a goal state, but is disturbed from the outside. The regulator samples the system output, compares to the goal state and determines the level of input.

![Figure 3 The classical model of a regulated system applied to transportation.](image)

A TCS is therefore a regulator that regulates a transportation system containing consignments. Disturbances can be expressed as the inherent complexity within the transportation system. TCS is thus defined:

*A TCS is a system that controls the trajectory of a transportation process.*

The TCS is in this definition analogous to the regulator in cybernetics. A TCS can employ resources to perform specific tasks during a trajectory, such as moving or handling goods. A TCS does not influence the initial state, or the goal state of the transportation system, only the trajectory between them.

The scope of the TCS encompasses the whole trajectory, from initial state to goal state. There is one important distinction, though: the TCS *never initiates the transportation process*. The initiative is always taken outside the (operative) control of the TCS. Since this thesis takes the perspective of the TCS, it means that there is no debate in this document on *why* goods need to be transported. The adopted perspective assumes that a transport is needed, and that a TCS has been assigned to facilitate it. Within the domain of the TCS lies therefore the various handling operations, the operations in the nodes, in the links, and the activities and resources associated therewith. “The system” means the transportation system *from the viewpoint of a TCS*.

In several instances in this thesis, the TCS simply means “forwarder” – meaning an actor in a forwarding role. The term *forwarder*, however, is ambiguous and is for that reason not used in this thesis other than to denote actual forwarding companies in case studies and examples.
1.2.4 The Consignment

The second unit of analysis, the *consignment*, is here defined as:

*A consignment consists of the physical objects that from an administrative point of view are treated as a single shipment and therefore - throughout a specific part of a supply chain - are given a unique identity.*

A consignment always has an initial state and a goal state that differ at least geographically. The consignment is hierarchically structured and may consist of sub-consignments on multiple levels (as can be seen in Woxenius, 1997; Woxenius, 1998, p. 101). Each sub-consignment carries its own attributes. The collective values of all the attributes constitute the consignment’s state.

A consignment does not “create itself”. It is created by a consignor. The consignor (or some other external actor) determines the ultimate goal state of the consignment as well as the initial state. Neither the consignment itself nor the TCS can alter these states.

1.2.5 Interaction between Transport Control System and Consignment

It is possible to study the relation between the TCS and a consignment from different perspectives.

From the perspective of the TCS, the consignment is an object that needs to be manipulated and processed through a transportation system so that it achieves the prescribed goal state.

If the perspective of the consignment is adopted, the TCS is manipulated in such a fashion that the consignment reaches its goal state. For every transport there are in fact several interactions between a consignment and one or more TCSs.

Note that the term “interaction” implies bi-directional influence. The traditional way of thinking may regard the consignment as a passive object that is “managed”. There are however several ways for a consignment to actually communicate with an external system. The current level of technology allows single products to carry active RFID tags, which enable them to make decisions and to act upon stimuli. Therefore, in anticipation of the future, the term interaction is used. A consignment that forces the current control system to recognise it as an individual is actually interacting with it.

1.2.6 The Interface

The interaction between Consignment and TCS is carried out through their respective *interfaces*.

There are several definitions of the term interface; most of them are contextual, i.e. they adhere to certain application areas (computer programming, chemistry, electrical engineering etc.). In this thesis a neutral definition is adopted:

*An interface is a representation of a system displaying its visible state and its visible inputs.*

“*Representation*” means that the interface is, in fact, a model of the system it represents. It is not equal to the system as such, but it *represents* the system relative to the observer.
“Display” (used instead of “contain”) means that the interface is viewed/accessed/manipulated by some other entity. The same system may thus display different interfaces to different viewers.

“Visible state” means that not all of the system’s attributes and their values may be visible through the interface. The interface is an abstraction, hiding the inner workings of the system. Throughout this thesis, the collection of visible attributes for a system is denoted its outbound interface.

Manipulating the state of a system needs input. “Visible inputs” represents a list of possible inputs to the system. Throughout this thesis, the collection of visible inputs for a system is denoted the inbound interface.

Consider a laptop computer. The screen, which contains the visible state (output), is the outbound interface. The keyboard, which is used for input is the inbound interface. Screen and keyboard both represent the computer, but in different ways.

To conform to the theories of object-orientation, the term operation is used to denote a displayed input for an interface.

1.2.7 Complexity drivers

The term complexity and its application on transportation and logistics has recently been extensively covered by several authors (Hultén, 1997; Franzén, 1999; Waidringer, 2001; Nilsson, 2003; Nilsson, 2005; Nilsson, 2006).

Waidringer defines transportation and logistics systems’ complexity:

“Transportation and logistics systems’ complexity resides in the nature of the network, process and stakeholders. It is a measure of the possibility of modelling these properties and their dynamic interaction in a way that allows of implementation of control mechanisms, forcing the system under study to meet required service, cost and environmental demands.”

(Waidringer, 2001, p. 48)

Waidringer’s division of the system into network, process, and stakeholders serves to emphasise different aspects of complexity. Waidringer emphasises the need for good models to be able to control the system. These models have to take into account not only the static relationships, but also the dynamic interactions between the parts of the system as well as its interaction with the environment.

Hultén proposes the use of cybernetics to address the complexity in logistics systems (Hultén, 1997). Cybernetics (as will be elaborated upon further in chapter 3) is a branch of systems science, and one of its key trademarks is that any system can be described to the outside world as an information processor – a Black Box. A black box transforms input into output and the goal in cybernetics is to control this black box so that the desired output is achieved in the most efficient way (Ashby, 1956; Beer, 1959).

To be able to distinguish one interaction from another, and to be able to compare them, a number of complexity drivers are studied. A complexity driver is a property of the studied system that increases the complexity of the total system.

Waidringer (2001) proposes three core properties of transportation and logistics systems that influence complexity: network, process, and stakeholder properties. Each of these represents a separate perspective on complexity. The Network perspective sees the system as a network where parts are interconnected to various degrees of topologic complexity. The Stakeholder perspective takes the viewpoint of a decision maker (a “stakeholder”) that faces a cognitive
complexity. The Process perspective envisions the system as a chain of events, whose complexity is called *algorithmic* (Waidringer, 2001).

Klir (1991) identifies three general principles in systems complexity, *descriptive, uncertainty-based, and computational complexity*. Descriptive complexity stems from the number of entities in the studied system and how much information is needed to describe them (variety). Uncertainty-based complexity is proportional to the amount of information that is needed to resolve any uncertainty in the system. The computational complexity is analogous to Waidringer’s algorithmic complexity (Klir, 1991, pp. 115-134).

Table 1 *Three complexity types, their measurements, and corresponding drivers.*

<table>
<thead>
<tr>
<th>Complexity type</th>
<th>Measurement</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptive (Klir)</td>
<td>Variety, connectivity, interdependence</td>
<td>Size of state space</td>
</tr>
<tr>
<td>Topologic (Waidringer)</td>
<td>Information needed to describe the system</td>
<td></td>
</tr>
<tr>
<td>Computational (Klir)</td>
<td>Uniqueness, knowledge, transformation size</td>
<td>Number and length of valid trajectories</td>
</tr>
<tr>
<td>Algorithmic (Waidringer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncertainty-based (Klir)</td>
<td>Entropy, information needed to resolve embedded uncertainty</td>
<td>Number of decisions that must be made during the transportation process</td>
</tr>
<tr>
<td>Cognitive (Waidringer)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Measuring complexity is difficult, not at least because the concept is so widely used and with various meanings.

Three principal drivers of the complexity have been identified. These drivers serve as gauges to measure the increase or decrease of complexity for a certain individual trajectory, such as a transport operation.

**Complexity driver: Size of state space**

*Variety* is defined as the number of distinguishable states in a system (Ashby, 1956). The more states a system can assume, the greater the variety. When regulating a system, the regulator must possess at least the same variety as the regulated system in order to be successful - The Law of Requisite Variety (Ashby, 1956).

Figure 4 below contains a network of nodes in a transportation system. There is a start node labelled Consignor and an end node labelled Consignee. There are ten additional nodes in the system, ranging from local road terminals to national hubs and ports. A transport of a consignment between consignor and consignee must traverse this network in order to reach its destination. The consignment may therefore be found in one of these twelve nodes. This means that the system depicted in Figure 4 has a variety of 12.
Figure 4 A transportation system consisting of 12 possible nodes (states). Goods are to be transported from the consignor to the consignee. To clarify the figure, the possible relations are also shown, although they are not part of the variety.

Waidringer explains connectivity in relation to what he calls topologic complexity in a transportation or logistics system, i.e. “... how the nodes and links in the network are connected and weighted” (Waidringer, 2001, p. 45). The more nodes and links in a system, the higher the complexity, especially if they are interdependent (i.e. that objects affect the states of each other).

Equation 1 The state of a system is described through attributes. The attributes can be discrete, i.e. a range of valid values, or continuous, i.e. within an accepted interval of values.

\[
\tilde{S} = \{a_1, a_2, ..., a_n\} \\
\mathcal{A}_i = \left\{a_{i,1}, a_{i,2}, ..., a_{i,m}\right\} \\
a_{i,1} \leq a_i \leq a_{i,2}
\]

In Equation 1 above, the complexity drivers is the number of attributes, \(n\), together with each attribute range, \(a_i\). The state space is comprised by all distinguishable variants of \(\tilde{S}\). Therefore, if the variants of \(\tilde{S}\) increase, so does the state space and so does the complexity. Variants of \(\tilde{S}\) can increase in two ways; increase the number of attributes in the system or increase the value ranges for the existing attributes.

Complexity driver: Number and length of valid trajectories

Heylighen (1996) argues that requisite variety is not enough, but that a successful regulator also must possess knowledge in order to choose correctly from the various alternatives that a large-variety system offers:
“In order to adequately compensate perturbations, a control system must "know" which action to select from the variety of available actions. This principle reminds us that a variety of actions is not sufficient for effective control, the system must be able to (vicariously) select an appropriate one. Without knowledge, the system would have to try out an action blindly, and the larger the variety of perturbations, the smaller the probability that this action would turn out to be adequate. Notice the tension between this law and the previous one: the more variety, the more difficult the selection to be made, and the more complex the requisite knowledge.”

(Heylighen, 1996)

Heylighen calls this the Law of Requisite Knowledge. This law highlights the fact that a good regulator needs to make decisions and that a high regulator variety is not necessarily sufficient for the successful, efficient state transformation.

Several studies have been made on the learning curve of systems where it is evident that the efficiency of the performance of new tasks improve over time, and also that infrequent exposure to unusual or new tasks decrease efficiency (Pattison and Teplitz, 1989; Synan and Larson, 1989; Adler and Clark, 1991; Boze, 1994; Almgren, 1999; Kontogiannis, 1999; Raggad and Gargano, 1999; Waterworth, 2000; Franceschini and Galetto, 2004; Rowe, 2005).

A TCS, which is required to control the transformation for a consignment where attributes are outside the normal ranges, or even the presence of new attributes, is subject to a learning curve; the efficiency by which the TCS is able to perform its task is dependent on the knowledge it has of the specific state change and its “best” trajectory. Without venturing into behavioural science it is assumed that any regulating system trying to manage complexity need training in order to perform its task to the best of its ability.

One important aspect is thus the knowledge of trajectories. Even though a system may have a large state space, there may be several of these states that are unattainable for practical reasons. It is part of the requisite knowledge to identify the valid states and to choose between them. Therefore, the number of valid states, i.e. the number of alternative trajectories that are available, directly affects the requisite knowledge and thereby the complexity.

In Figure 5, below, the same system that was shown in Figure 4 now contains directed links between the twelve nodes. The links together with the nodes form a directed graph that depicts every possible trajectory in the system.
Figure 5 In this picture, the transport network is regarded as a simple directed graph. There are several possible trajectories between consignor and consignee, depending on which forwarder is chosen and which mode of transport.

Also, the length of a trajectory affects the requisite knowledge. When planning a transportation process, the choice between alternative trajectories becomes more difficult if the trajectories themselves increase in complexity.

Complexity driver: Number of decisions that must be made during the transportation process

One of the most common suggestions for measurement of complexity is the information entropy (Shannon and Weaver, 1949; Beer, 1959; Wiener, 1961; Hultén, 1997). In information theory, entropy is the expected amount of information that is needed to describe the state of a system (Sivadasan et al., 2002). This measurement focuses on the structure of the system at a certain point in time and correlates with the uncertainty-based complexity (Klir, 1991 p. 117).

For transportation systems, entropy is defined by Hultén as:

“… a measure for the indeterminateness of a system. The indeterminateness can be in terms of lack of specification in a description of the system or in terms of an inherent indeterminateness of the state of the system.”

(Hultén, 1997, p. 40)

Waidringer defines the cognitive complexity for logistics systems:

“… the logistics system’s complexity is high when the stakeholders involved in the system find it difficult to comprehend the totality of the system.”

(Waidringer, 2001, p 62)
Klir, Hultén, and Waidringer all support the notion that an incomplete image of the system increases complexity. Hultén and Klir present a measurement for this incompleteness: entropy. A TCS controlling a trajectory is dependent on the quality of its “map” of the controlled system. This map is, compared to the true system, an abstraction – a simplification – of the system and will therefore hide variety from the TCS. Hultén writes:

“The map of the system can be more or less detailed. Entropy is a measure for the indeterminability of the system with our present map of it. When we obtain more information the uncertainty about the current state of the real system decreases.”

(Hultén, 1997, p. 42)

In other words: the better the model, the less uncertainty. The best regulator is in fact, according to Conant and Ashby (1970), a model of the system it regulates. They also state that a successful regulator, which is not a model of the system it regulates, is “unnecessarily complex” and that the variety in the regulator can, in fact, be too high (Conant and Ashby, 1970).

In Figure 6 below, the modelled system from Figure 4 and Figure 5 lacks information on several links, presenting an incomplete image of the system. A TCS possessing such a model of a system needs to acquire information on the missing elements in the model before decisions can be made.

![Figure 6](image-url)

Figure 6 In this picture, the model of the network is incomplete. It will not be possible to plot a trajectory until more nodes and/or links are known, i.e. the entropy is diminished.

Uncertainty can be handled in two ways: influence the system to assume the state that the model is currently showing, or change the model to reflect the true state of the system (Hultén, 1997). Both of these strategies involve “stepping into” the system and perform either an operation to change the system state or a remodelling to more accurately depict the current state. In the first strategy, the inbound interface needs to be redesigned since the needed
operation is not accessible to the TCS. In the second strategy, the outbound interface is inadequate in describing the true state of the system and needs to be remodelled. Each of the strategies implies that decisions must be made by the TCS during the actual transportation process. These decisions cannot be modelled beforehand, since they pertain to uncertainties that arise due to inadequate modelling. Also, these decisions cannot be made “within” the TCS since the TCS lacks the information that is needed in order to make them. Therefore, if the number of decisions that need to be made during the trajectory increases, so does the complexity.

1.2.8 Research questions

The complexity drivers (Chapter 1.2.7) together with the TCS (Chapter 1.2.3), the Consignment (Chapter 1.2.4), and the Interface (Chapter 1.2.6) comprise the focus of the following research questions.

The questions are phrased to trigger the investigation into the actual interactions between TCS and consignment (see Chapter 1.2.5) with regard to the complexity of the transportation system.

Q1 How does the interface of the consignment affect the complexity drivers during a transportation process?

Q2 How does the interface of the Transport Control System affect the complexity drivers during a transportation process?

Q3 How should the interface between the consignment and its Transport Control System be designed in order to minimise the impact of complexity in the control of the transportation system?

The complexity is a measurement of how “demanding” a consignment is of the TCS during its trajectory. The first two questions focus on the study of the complexity drivers, and how these are affected by changes in the transportation system such as changes in the interfaces of either consignments or TCSs. The third question takes a normative approach and seeks an answer to how consignment and TCS can be designed in order to minimise the effects that the complexity has on the transportation system.

Q1: How does the interface of the consignment affect the complexity drivers during a transportation process?

A consignment is transported, handled and otherwise manipulated to achieve its goal state. How the changes of attribute values are performed – in what way and in what order – determines the amount of resources that is needed.

The outbound interface of a consignment, i.e. its attributes, describes its state at a certain point in time. The data structure of the outbound interface (the number of attributes and their value ranges) determine every conceivable state that together form the state space for the consignment. The driver behind the descriptive complexity is the state space, which in this case is represented by the data structure of the outbound interface of the consignment. Therefore, in order to study changes in descriptive complexity, the changes in the outbound interface must be studied.

The computational complexity depends on the number of valid trajectories (and their lengths) of the consignment. Several of the states, which are theoretically possible for a consignment to assume, are not practically feasible. Also, there may be sequential restrictions on the state transitions so that a certain state can only be reached from a few other states. There are therefore several restrictions that limit the number of valid trajectories for a certain consignment. The computational complexity depends on how difficult it is for the TCS to
a) identify the valid trajectories and b) choose the best trajectory according to some predefined criteria.

A consignment has a goal state already predefined when a TCS assumes control. It is the task of the TCS to control the consignment so that this goal state is reached. There is always a certain amount of uncertainty embedded in a transportation system. The system contains several interacting subsystems that form complex structure of interdependence. For a consignment, this uncertainty can bring difficulty for a TCS to, in advance, plan the exact trajectory that is needed to reach the goal state. There are for instance choices that depend on the states of subsystems and, therefore, cannot be made in advance because these states are not known. A driver behind this uncertainty-based complexity is the number of decisions that cannot be made when planning a trajectory, but rather has to be made during the transportation process.

Q2: How does the interface of the Transport Control System affect the complexity drivers during a transportation process?

The design of a TCS has direct bearing on what types of consignments it can control, and how well it can calculate and facilitate their trajectories. This second research question seeks an answer to how this design affects the complexity drivers during a transportation process.

The descriptive complexity for a transportation process can be affected by the TCS design. Since the TCS is regarded as a regulator that controls the trajectory of a consignment, it is subject to the Law of Requisite Variety (see page 8), stating that the variety of the TCS must be equal to (or greater than) the variety (state space) of the consignment. Therefore, for the TCS, the variety has a lower boundary defined by the consignments it controls. The focus in this research question regarding the descriptive complexity will thus be on how the appropriate variety is reached through the design of the TCS interface.

When controlling a consignment through a transportation process, the TCS needs to choose a trajectory. This choice, as was explained in the previous section, depends on how many alternative trajectories there are and how they differ from each other. The task of the TCS is to identify the valid trajectories and then choose between them. This research question will examine how this choice may be affected by the interface design of the TCS. The computational complexity is an indication on how difficult it will be for the TCS to decide upon a trajectory for a consignment.

Uncertainty-based complexity is based on the TCS having an incomplete model of the transportation system it controls. Uncertainty affects the TCS’s ability to perceive the state space as well as the alternative trajectories. During a transportation process, the TCS may need to make decisions regarding a specific trajectory based on the fact that information needed to make these decisions was not available to the TCS before the transportation process began. This research question focuses both on the TCS and on the effect its interface design has on the number of decisions that are made during transportation processes.

Q3: How should the interface between the consignment and its Transport Control System be designed in order to minimise the impact of complexity in the control of the transportation system?

When designing a transportation system, several important decisions are focused on the design of interfaces; both the TCS interfaces and the desired consignment interfaces.

This research question introduces a normative aspect on how interfaces should be designed to minimise the impact of complexity when controlling a transportation process. Research questions Q1 and Q2 address the mechanisms behind the complexity and how they are affected by interface design. Q3 has the ambition to suggest design principles and
implementation strategies for future systems with complexity reduction in focus. It is difficult to measure the complexity directly, hence the identification of complexity drivers that may facilitate an indirect measurement. Therefore, for this research question, the search is for the best way to design a system from the perspective of the identified complexity drivers.

1.3 Structure of thesis

This section describes the different parts of the thesis and how they relate to each other. There are seven chapters, including this one (see Figure 7 for a graphical representation of the thesis).

Chapter 1 addresses the aim and scope of the research as well as the formulation of research questions. First, the basic problem is specified. Within this problem lie the various system operations that are performed to meet requirements posed by a consignment. This leads to, with a focus on the forwarder’s problems, the formulation of three research questions. These three questions are then referred to from different angles throughout the thesis. Chapter 1 is supplemented with an Appendix containing a list of terms.

Chapter 2 elaborates upon the methodology and how the results from the empirical data collection will be analysed. This thesis takes a systems perspective. This means that there is limited possibility (and limited desire) to strive for repeatability. The very core of systems science is that a reductionist approach is doomed to fail when applied to describe and explain a system that displays emergent properties.

Instead, a creative approach is needed: In the description of a system, the researcher partly unknowingly chooses what information is interesting and what information is not. This abstraction is in itself a product of the researcher’s preconceptions and knowledge about the subject and is therefore an important component in the overall data analysis. Chapter 2 describes the methodological aspirations for this thesis and also elaborates upon the different tools that are used in the processing and analysis of the data.

Chapter 3 contains the frame of reference. The theories of transportation help define the problem domain and where to draw the system boundaries. The area of risk analysis theory is used when defining Goods Requiring Special Attention, itself a building block for this thesis. Systems theory facilitates the use of several well-defined models, such as the Black Box. Object-orientation (OO) is a powerful modelling philosophy that enables several views of a system, describing class- and object hierarchies as well as specific cases and generic rules. OO is applied systems science with close ties to the concepts of cybernetics, although with a sophisticated visualisation scheme not found in cybernetics.

Chapter 3 is supplemented with three appendices. Appendix 2 contains a basic description on object-orientation, about its concepts and tools. Appendix 3 contains a reference on the Unified Modeling Language (UML) that is used to construct the case models. Appendix 4 contains a list of papers reviewed in Chapter 3.5.

Chapter 4 contains deeper discussions on the theories from the frame of reference and how they can be extended to encompass the problem domain of this thesis. Although object-orientation can be considered “applied systems science”, some aspects of it may, however, still be considered as outside the scope of systems science, such as the activity of programming. The notion of transportation as a systems science is also subject of discussion. In this thesis, a systems perspective is adopted, although a transportation chain may be regarded with for instance a mathematical and quantitative approach, e.g. the Travelling Salesman problem. These mathematical methods, often used by operations researchers, serve to solve a different set of problems than those associated with the systems perspective. Also, object-orientation is often used when applying quantitative methods to transportation systems. Object-oriented modelling is an excellent tool to use when designing large optimisation
problems, since the construction of the algorithms closely resembles programming. The triple overlap between the areas of systems science, transportation, and object-orientation is the theoretical focus in this thesis. Chapter 4 discusses the various aspects of object-oriented systems models of transportation chains.

In Chapter 5, the eight cases are presented. Each case is described in a separate section. Each of the separate cases has been chosen because they represent different parts of the same hypothetical trajectory.

Chapter 6 contains analyses of the empirical data from the cases coupled with the theory within the frame of reference. First, a class library is constructed from the cases together with object-oriented models of each case (supplemented in Appendix 5). Second, a cross-case comparison is conducted on the first two research questions. Third, a calculation example is presented where data from one of the cases is used to construct an object-oriented model that can be manipulated in a simulation program, calculating results based upon changes in the model. Data on the calculation is supplemented in Appendix 6. Chapter 6 concludes with a section examining the research as such in terms of validity and reliability.

Chapter 7 addresses the third research question and concludes the thesis. In Chapter 6, the various aspects of control were examined. Chapter 7 focuses on the long-term control, which mainly consists of design activities. In these, a Transport Control System can, in advance and long before any consignments arrive, define the outer limits of the transportation system. The long-term control has the power to reduce not only the descriptive complexity, but also the computational and uncertainty-based. The descriptive complexity is the easiest to reduce in the long-term control scope. This chapter also addresses the issues of designing for heterogeneous goods and the uses of object-oriented modelling in logistics and transportation.
Figure 7: Structure of the thesis.
2. METHODOLOGY

The approach to methodology is very much a product of paradigm. This thesis takes a systems perspective. This means that there is limited possibility (and limited desire) to strive for repeatability. The very core of systems science is that a reductionist approach is doomed to fail when applied to describe and explain a system that displays emergent properties.

Instead, a creative approach is needed: In the description of a system, the researcher partly unknowingly chooses what information is interesting and what information is not. This abstraction is in itself a product of the researcher’s preconceptions and knowledge about the subject and is therefore an important component in the overall data analysis. This chapter will describe the methodological aspirations for this thesis and will also elaborate upon the different tools that are used in the processing and analysis of the data.

2.1 Knowledge creation

In essence, the three research questions in this report are merely signposts pointing towards a larger goal. If answered in a comprehensive enough manner, the questions will contribute to the collective body of theoretical knowledge on goods type-related effects in transportation systems. The contributions are not in the form of new formulas, but rather in the forms of new ways of describing certain aspects of a studied system. In describing the system differently than what has been done previously, new understanding may emerge. In the longer term, a new way of describing a system may in some way actually affect the system itself, mainly because a new “description language” highlights different aspects of the already known system. Brunsson writes:

“There are several ways to change social systems. /.../ [Science] affect [systems] by creating ‘languages’ to describe and understand various situations and by communicating these languages to the actors of the systems.”

(translated from Brunsson, 1982, p. 106)

Thus, according to Brunsson, a change in a system’s language actually can change the system itself. One of the objectives of science is to study a phenomenon, gain understanding of how it works and then, when possible, use this knowledge to improve or otherwise affect the phenomenon. Therefore, when creating “languages” in systems science, it is important to bear in mind the fact that the created language may affect the studied system.
In this thesis, the following guidelines have been used when creating the language used in models and descriptions:

- **Use the right metaphors.** New theory does not have to be incomprehensible to anyone not having an academic degree. The choice of metaphors is critical to gaining understanding outside the academic domain.

- **Strive to affect the language used in transportation systems to describe and control their operations.** By advocating a shift of focus through the use of language, an entire system may change. Examples may for instance be found in the public sector where concepts such as customer and service provider are introduced in medical care institutions. When a patient becomes a customer, new mechanisms of control and management emerge and new behaviour can be seen (Walters and Jones, 2001).

Since the objects of study in this research are transportation systems, the language used in modelling etc. has been intended for professionals within this field as well as for the academic community.

### 2.1.1 Grounded theory

The creation of new theory is not a straightforward process. The ideal deductive case would be to first formulate a theory, then to test that theory with empirical data. The opposite, an ideal inductive approach, starts with collection of data which leads to analysis and building of theory from that data. In reality, a more mixed approach is often the case. The results of this thesis are mainly based on inductive research, although not on “pure” induction. The gathering of data has been more or less deliberate towards a certain goal, even though the theory as a whole has evolved during the process rather than having been complete from the beginning. Grounded theory, i.e. theory based on empirical data, is a widely used label for research that is mainly inductive. Glaser and Strauss explain:

“/…/ grounded theory is derived from data and then illustrated by characteristic examples of data.”

(Glaser and Strauss, 1968, p. 5)

The uses of grounded theory are numerous, as are the pitfalls. When working closely with the data during a case study, for instance during interviews, the researcher must be careful not to draw premature conclusions or to filter out data that does not correlate with a preconceived theory.

“A researcher does not begin a project with a preconceived theory in mind (unless his or her purpose is to elaborate and extend existing theory). Rather, the researcher begins with an area of study and allows the theory to emerge from the data.”

(Strauss and Corbin, 1998, p. 12)

In this research, there was a preconceived theory, formulated in Arnäs (1999a), regarding the properties of heterogeneous goods that affect transportation cost. Here, that theory is extended, partly based on findings in case studies, partly based on theoretical research.
When collecting data, it is the data that will provide the theory and not the other way round. If a researcher enters the field with a preconceived theory he can easily find examples supporting that theory. But, as Glaser and Strauss says:

“/…/ since the idea has not been derived from the example, seldom can the example correct or change it (even if the author is willing), since the example was selectively chosen for its confirming power.”

(Glaser and Strauss, 1968, p. 5)

In the case of this thesis, there is a danger of unintentional bias due to erroneous preconceptions on how the complexity generated by heterogeneous goods types changes. For instance, the example stated in the introduction to the thesis (Chapter 1.1) is no more than an isolated incident, and may well be a single event that will never repeat itself. It may also be an indication of a pattern that emerges as more and more studies are made. This emergence needs critical mass, in this case a critical amount of data, before patterns become discernable.  

The three research questions are formulated with the intent of delving into the actual interactions that take place between Consignment and TCS. These interactions may vary in complexity and frequency and cannot be considered homogeneous. A large number of interactions need to be studied in order to address the research questions. It is inevitable that when performing a large number of studies, consisting of several interviews each, the researcher starts to process the data during the collection process. This processing is an integral part of grounded theory and allows reshaping of the study as new patterns emerge. Strauss and Corbin explain the importance of allowing emergence and to be aware that such emergence occurs (Strauss and Corbin, 1998, p. 34).

When emergence is allowed and acted upon, the data collection process actually co-evolves with the theory. It is important that the researcher keeps focus in this stage. Even though the theory is evolving in perhaps unforeseen directions, there still has to be a connection to the original objectives of the study. After all, exploration is only fruitful if the explorer actually returns with the results. Even so, Strauss and Corbin state that the theoretical sampling is likely to mostly be the effect of happenstance rather than planning:

“Unlike statistical sampling, theoretical sampling cannot be planned before embarking on a study. The specific sampling decisions evolve during the research process. Of course, prior to beginning the investigation, a researcher can reason that events are likely to be found at certain sites and populations.”

(Strauss and Corbin, 1998, p. 215)

This does not mean that a researcher is a slave under the collected data. During most studies, more than one pattern may emerge, and each pattern may prompt several likely continuations in the data collection. Mostly, there are several options during a study. One way to strategise is to search for “likeness”. If a certain phenomenon is observed in one case, the return to a previous case, searching for the same phenomenon there may be fruitful. The more connections of this type that can be made between seemingly disparate data sets, the more credibility is lent to the grounded theory that may be the result of these connections. Strauss and Corbin stress the need for “density” of properties and dimensions (Strauss and Corbin, 1998, p. 201).

---

4 The term “emergence” is also used in systems science, with the meaning that a system displays emergent properties as a whole that cannot be found when studying its parts.
In this thesis, the theory that is created is divided into two separate parts. First, there is Chapter 4, “Building a framework”, which is a typical desk research product where various theories are combined and synthesised into new models and constructs. Chapter 5, “Case studies”, provide the data that will assist in generating the second part of theory. To this end, the analysis (Chapter 6) combines the empirical data with the theoretical results of Chapter 4.

2.1.2 The power of language and metaphor

Morgan (1980) adopts a hierarchical view of how paradigms influence research activities. He states that each paradigm provides a number of metaphors that form the basis of the “schools of thought” within each paradigm. Each metaphor, in turn, gives birth to a number of specific “puzzle-solving activities”.

![Figure 8 Paradigms, metaphors and puzzle-solving. Three concepts for understanding the nature and organisation of social science, according to Morgan (redrawn from Morgan, 1980).](image)

This hierarchic view implies that each metaphor contains exclusive puzzle-solving activities. This is of course not always the case since a certain activity can be used to support more than one metaphor.

2.1.3 Paradigms

The term paradigm was first introduced by Kuhn in 1962. He defined it:

“Men whose research is based on shared paradigms are committed to the same rules and standards for scientific practice.”

(Kuhn, 1970, p. 11)

Since then, the term has been redefined and clarified further by several scientists. Morgan describes a paradigm as an alternative reality that can be broken down into metaphors being used to clarify and define the nature of this reality. Arbnor and Bjerke (1997) discuss the relation between what they call paradigm and operative paradigm. The researcher’s ultimate presumptions form a paradigm, which in turn leads to the operative paradigm (comparable to the metaphors in Morgan’s model in Figure 8 above).
Figure 9 Arbnor’s and Bjerke’s model of how a paradigm and an operative paradigm respectively influences research methodology (redrawn from Arbnor and Bjerke, 1997, p. 17).

Arbnor and Bjerke relate methodology to the construction of methods.

“Methodology is the understanding of how methods are constructed, that is, how an operative paradigm is developed.”

(Arbnor and Bjerke, 1997, p. 16)

In the table below, the model from Figure 9 is adopted for this thesis.

Table 2 In this thesis, the paradigm and operative paradigm are grounded in systems science.

<table>
<thead>
<tr>
<th>Ultimate presumptions</th>
<th>The Universe is tangible and can be studied objectively.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory of science</td>
<td>The Universe displays systemic properties, as well as any subset thereof that consists of more than one distinct, interacting, part. This systemic behaviour cannot be fully understood from within the system.</td>
</tr>
<tr>
<td>Methodological approach</td>
<td>Systems science</td>
</tr>
<tr>
<td>Methodology</td>
<td>Cybernetics and object-oriented methods</td>
</tr>
<tr>
<td>Study area</td>
<td>Transportation systems</td>
</tr>
</tbody>
</table>

This thesis takes a systems perspective. The systems paradigm is based upon the notion that a set of elements that are ordered, or in some other way interact, display emergent properties. These emergent properties are specific to the whole as a single unit, not to the parts. The systems perspective is elaborated upon further in Chapter 3.2.
The method of reducing a study object into manageable parts, which can be “solved” or studied separately (also called reductionism), is not applicable in systems research. There is a consensus in the systems research area regarding some of the characteristics of a system (see for instance Ashby, 1958; Beer, 1959; Ackoff, 1973; Checkland, 1976; Gougen and Varela, 1979; Churchman, 1981; Checkland and Scholes, 1990):

- A system is composed of more than one (elementary) part
  - There are always what can be called elementary parts, building blocks, which are not destroyed or divided into other parts within the time horizon that the system is studied.
  - For something to be called a system, it must contain at least one set of such parts. The set must contain at least two parts.

- A system differs from a set when it displays emergent properties
  - When a set displays some function or property that is related to the whole, this property is called emergent property. The set is then a system.

Cybernetics is an area within systems science that focuses on the notion of control, or steermanship (the word cybernetics is derived from the Greek word kybernetes meaning steersman). To adopt cybernetics as a methodology means that the following stance is taken (see for instance Wiener, 1961; Ashby, 1956; Beer, 1959; Klir, 1991):

- Control focus
  - How is the studied system controlled?
  - How can the studied system be controlled?

- Input-output focus
  - A cybernetic model centres around the notion of input and output
  - Input and output makes possible the regulation (or control) of the system

Object-orientation is first and foremost a modelling philosophy where several pre-existing modelling methods have been integrated into a coherent framework. Object-oriented modelling is primarily used in software engineering, but has recently surfaced as an all-purpose modelling framework for arbitrary systems. This thesis proposes that the object-oriented framework is aptly suited for cybernetic modelling and that the powerful visualisation techniques, which are part of object-orientation, provide meaningful additions to cybernetic modelling. On object-orientation, see for instance Appendices 2 and 3 or Booch (1991), Rumbaugh (1991), Yourdon (1994), or Larman (1998).

2.1.4 Metaphors

A conscious use of metaphors can be very effective in describing and studying a system. Also, unconscious use may be ineffective and even damaging. It is therefore of utmost importance that the metaphors that are used are done so by a conscious intent, and not “by accident”. Morgan (1980) discusses the uses of metaphors in puzzle-solving. Morgan writes:

“The use of a metaphor serves to generate an image for studying a subject. This image can provide the basis for detailed scientific research based upon attempts to discover the extent to which features of the metaphor are found in the subject of inquiry.”

(Morgan, 1980, p. 611)
A metaphor can thus be a very powerful tool when studying a system. The system sciences are relatively “metaphor-dense”.

The models that are used in transportation and logistics often refer to concepts outside the area. Consider modelling information in a transportation system. There are at least two conflicting metaphors:

- **Information flow**
  - Information is moved; it can be transmitted and received.
  - The use of the flow metaphor implies movement, dynamics, flow rate, sequence, etc.

- **Information web**
  - Information is accessed, it is either available or not.
  - The use of this metaphor implies multiple access points, that changes are simultaneously propagating, absence of definitive boundary, etc.

Both of these metaphors are useful. The flow metaphor can be used to study the way a transportation system behaves in a specific situation or in a specific case. This metaphor does not quite explain the uses of an order database in a forwarding company, however. The notion of a flow is apparently not applicable to some kinds of information. The web metaphor, on the other hand, is proficient in explaining the ordering system, but is found wanting when applied to a single case. It is therefore crucial to choose the right metaphors.

The trademark of a good metaphor is that there are just enough similarities between the metaphor and the studied subject. They can neither be too separate, nor too similar. It is for example difficult to envision a room full of schoolchildren as a model of a transportation system. A 1:20 scale model of the studied transportation system might be useful when studying infrastructure, but not when studying the effects of differences in goods type. Cornelissen (2003) emphasises that an apt metaphor needs to represent the intended system heuristically:

> “The validity of metaphors in academic language comes thus from their capacity of serving a valid heuristic purpose: their truth or falsity is, depending upon the context, the central criterion for judging whether or not they are apt and fitting to the subject under consideration.”

(Cornelissen, 2003, p. 222)

In order to find the right type of metaphor that is useful for this project, the following conditions must therefore be met:

- A transportation process must be implied. There needs to be a way for the metaphor to indicate transformations in space.
- A way to study the cost of a transport (in any unit) is needed, and will be required of a metaphor. The “cost” for the transformation must also be expressed in the metaphor.
- A metaphor must allow differences in goods type to be studied.

In this thesis, object-oriented modelling is used to provide a number of metaphors for a transportation system. These metaphors are more or less abstract, and serve as ways to efficiently model and visualise a transportation system from several viewpoints.
2.1.5 Puzzle-solving activities

Puzzle-solving activities include the various tools that are used to achieve a puzzle solution. In this thesis the solution needed is the answer to the three research questions in line with the aim as stated in Chapter 1.2.1.

The tools at disposal are, according to Morgan, embedded in the paradigm and come as results of the metaphors that are used. In the case of this thesis, some metaphors are taken from systems science (specifically cybernetics). The tools available for studies of cybernetic systems are numerous. Depending on the type of system, the tools vary from purely mathematical to loosely defined case studies. When studying a cybernetic system, it is imperative to enumerate as many attributes as possible. The attributes are treated as variables, often in a pure mathematical sense. The idea behind the cybernetic system, however, is not necessary quantifiable.

Typical “puzzles” that can be addressed with a systems perspective include:

- Describing and understanding complex system behaviour. Multiple possible outcomes from a single input.
- Describing and understanding systems dense with human activities. As soon as the concepts of free will and personal initiative are introduced into a model, a system’s methodology of some degree is needed.
- Finding non-continuous cause-and-effect relationships. In the study of a system, the researcher forms chains of evidence that are not readily available through for instance mathematical methods.

The puzzles in this thesis are focused on how different objects (consignments) affect their control systems (TCSs) through their interfaces, and how the design of these interfaces can alter this effect.

2.2 Tools

In this section, the methodological tools upon which this thesis is based are presented.

2.2.1 Case study research

Case studies can be used in different situations. According to Yin, the case study can be used for three purposes: exploratory, descriptive, and explanatory (Yin, 1994). Eisenhardt states that case study research can be used to provide description, test theory, or generate theory (Eisenhardt, 1989).

As stated earlier in section 2.1.1, the results of this thesis are theories based on empirical data. The case study as a method has been widely used but it is important to emphasise that case studies are not always the “best” way to gain knowledge about a phenomenon. Depending on how the object of interest behaves, as well as the nature of the inquest, case studies will be more or less appropriate. Yin states in his now famous matrix on research strategies, that the case study is relevant when contemporary events, which are outside the researcher’s control, are studied. The phrasing of the research question may also indicate whether a case study is preferable to, for instance, a survey. If a research question or objective is more bent on understanding than enumerating and counting, then a case study will likely be appropriate.
Table 3 Relevant situations for different research strategies (redrawn from Yin, 1994)

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Form of research question</th>
<th>Requires control over behavioural events?</th>
<th>Focuses on contemporary events?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment</td>
<td>how, why</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Survey</td>
<td>who, what, where, how many, how much</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Archival analysis</td>
<td>who, what, where, how many, how much</td>
<td>No</td>
<td>Yes/ No</td>
</tr>
<tr>
<td>History</td>
<td>how, why</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Case study</td>
<td>how, why</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In this thesis, case studies have been used to collect the data that is used in addressing research questions 1 and 2 (what attributes affect TCS and consignment respectively). In the essence of the research questions is the query of how/why the complexity increases.

Yin provides what he calls a “technical definition” of a case study.

- **A case study is an empirical inquiry that**
  - investigates a contemporary phenomenon within its real-life context, especially when
  - the boundaries between phenomenon and context are not clearly evident

/.../

- **The case study inquiry**
  - copes with the technically distinctive situation in which there may be many more variables of interest than data points, and as one result
  - relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result
  - benefits from the prior development of theoretical propositions to guide data collection and analysis

(Yin, 1994, p 13)

As Strauss and Corbin state, when elaborating or extending an existing theory, the grounded theory approach can be used (quoted on page 20). In the last of Yin’s statements above, he states that the case study actually benefits from such a preconceived model.
It is important to distinguish between types of evidence (e.g. qualitative data), types of data collection methods (e.g. interviews), and research strategies (e.g. case studies). A case study may well rely solely on quantitative data as well as qualitative. Likewise is the choice of data collection method equally free of restraints. A case study represents a research strategy “to be likened to an experiment, a history, or a simulation, which may be considered alternative research strategies” (Yin, 1981, p. 59).

Selection of cases

Eisenhardt stresses the importance of case selection and says that “as in hypothesis-testing research, the concept of a population is crucial, because the population defines the set of entities from which the research sample is to be drawn” (Eisenhardt, 1989, pp. 536-537). This statement is supported by Strauss and Corbin (Strauss and Corbin, 1998, p. 201). There are several ways of defining the population (Eisenhardt, 1989):

a) Replicate previous cases or extend emergent theory. The cases are chosen because they fit a known system or a known theory.

b) Polar types with transparently observable processes, i.e. compare opposites using the same, common, variables.

c) Random selection. Select the cases without any preconceptions.

The selected cases in this thesis have been of types a) and b). The structure of the systems chosen for study has been known beforehand (type a), but several key attributes of them have been chosen to gain a coverage of as many polar types as possible (type b).

The number of cases that is needed depends on when theoretical saturation is reached, i.e. when new data does not contribute to new learning. Eisenhart actually recommends the number of cases to between 4 and 10 (Eisenhardt, 1989). This statement has been criticised by Dyer and Wilkins who point out that several well known and successful case studies consist of only one or two cases (Dyer and Wilkins, 1991).

In this thesis, since interactions between consignment and TCS are studied, several cases are needed to gain a decent population of not just consignments but also TCSs.

2.2.2 Object-oriented modelling

As already stated, object-orientation will be used as a modelling tool for transportation systems. The methodological foundations of object-orientation stem from computing as well as systems science (described in more detail in chapter 3.4.3). A review on the uses of object-orientation within transportation and logistics can be found in Chapter 3.5 on page 72.

The method is well established in computer science as a way to efficiently design transparent and maintainable computer applications (Yourdon and Argila, 1996; Yourdon, 1994; Rumbaugh et al., 1991; Jia, 2000; Brown, 1997; Booch, 1991). The method has more uses than that, however, since it can be used as a general modelling tool for just about any system (Object Management Group, 2001; Arnäs, 2001a). Also, object-orientation is a potent visualisation framework, where several standardised diagram types can be used to model states, relationships, processes, use cases, or trajectories.
There are a number of detailed methods available for object-oriented modelling and analysis, such as Booch, Jacobson, UML, or Coad/Yourdon.

Regardless of method, the basic principles of object-orientation are the same, even though the notation and the diagram types change somewhat. UML (Unified Modeling Language) has been developed by the Object Management Group (consisting of, among others, the people behind other the other methods listed above). The Object Management Group (OMG) has been working on a modelling standard since 1989. In the Object-Oriented community today (2007), UML is the closest thing to a standard there is and will therefore be used in this thesis (a brief description of UML can be found in Appendix 3).

By using the object-oriented modelling technique, a family of analysis tools becomes available. Object-oriented analysis (OOA) is a well-documented way of analysing a system consisting of an object model. OOD (object-oriented design) is used to construct models of future systems based on the analysis performed in OOA. OOA and OOD are both powerful and versatile tools that can be used in a number of different ways when analysing, studying and designing systems.

2.3 Research design

This section describes the strategy of the research. The research design is given in three stages.

The first stage, data collection, describes the strategy behind the collection of empirical data, and how interview objects are selected.

The second stage, data analysis, describes how the collected data will be processed and what information is needed from the data collection in order to address the research questions.

The third stage, design of future systems, contains a synthesis of the data analysis primarily aimed at the third research question.

The research questions are taken into account when setting up the strategy for data collection. In order to facilitate a good analysis, each question demands certain data to be collected.

Q1 How does the interface of the consignment affect the complexity drivers during a transportation process?

Q2 How does the interface of the Transport Control System affect the complexity drivers during a transportation process?

Q3 How should the interface between the consignment and its Transport Control System be designed in order to minimise the impact of complexity in the control of the transportation system?

The analysis will focus on how and why complexity increases. The cases provide data from different companies, different organisational levels, different products etc. All this data must be structured in order for an analysis to be possible. Focus on the research questions gives the following requirements:

---

5 The use of the term “method” means in this case the various modelling “dialects” that exist within the OO community.
Table 4 Requirements for the data gathering process in order to be able to perform an analysis.

<table>
<thead>
<tr>
<th>Unit of analysis</th>
<th>Information needed</th>
<th>Impact on analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each identified TCS within each case.</td>
<td>What distinguishes this particular TCS from others?</td>
<td>By differentiating the identified TCSs, the complexity drivers may be identified.</td>
</tr>
<tr>
<td></td>
<td>What are the typical operational procedures?</td>
<td>Gain further depth in the information on the interactions for each TCS.</td>
</tr>
<tr>
<td>Each identified consignment within each case.</td>
<td>What distinguishes this consignment from others?</td>
<td>By differentiating the identified consignments, the complexity drivers may be identified.</td>
</tr>
<tr>
<td></td>
<td>Was this consignment considered individually from other consignments?</td>
<td>If the consignment is treated as an individual, a direct interaction occurs.</td>
</tr>
<tr>
<td>Each identified interaction between TCS and consignment for each case.</td>
<td>What is the size of the state space?</td>
<td>By identifying the complexity drivers together with the differentiation of consignment and TCS, a pattern of interaction change as a function of TCS and consignment may emerge.</td>
</tr>
<tr>
<td></td>
<td>What is the number and length of valid trajectories?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>What is the number of decisions that must be made during the transportation process?</td>
<td></td>
</tr>
</tbody>
</table>

The table above gives only the required data for analysis, not the form in which this data is to be displayed.

2.3.1 Stage one: Data collection

Yin states that

“Evidence from case studies may come from six sources: documents, archival records, interviews, direct observation, participant-observation, and physical artifacts.”

(Yin, 1994, p. 78)

In the case studies performed for this thesis, the primary evidence sources are interviews combined with direct observations. The strategy has been to perform flow-oriented interviews and observations with as many people as practically possible. Since the collection of data has been largely exploratory, as is common in case studies, there has been no fixed questions or formulas to direct the data gathering (Yin, 1994, p. 84).
It is of great importance how the interview objects are selected. Any person in an organisation abstracts from his/her viewpoint. A transport is a pure information transaction to anyone not participating in the physical handling itself. It is therefore inevitable that persons, being removed from the events and phenomena they describe, rationalise and abstract from them. If several interview objects in a company were to be asked to describe the same procedure, there will be discrepancies just because of their differences in abstraction. Interview objects are model-builders of their environment. It can be safe to assume that they are the most proficient in describing their own work, and that they tend to become more abstract the further away they get from that topic. This does not mean that their views on other topics are not interesting. It can be very enlightening to ask interview objects about anything within the scope of the study to see where the discrepancies in the system are. Because of the limitations to these fields of vision, it is crucial to select interview objects with great care. Depending on who is selected, the descriptions may vary greatly. When performing the interviews for this thesis, the main strategy has been to follow the studied goods- and information flows as closely as possible. The selection of interview objects has therefore been made as the studied flows progress through the study.

With the first cases, the focus of the research was aimed at dangerous goods transport. When something goes wrong with a dangerous goods transport, it often means that a law is broken. It was therefore crucial that the interview objects felt comfortable with speaking of these things without the risk of exposure, neither of their identity nor the company’s. Several of the cases are thus anonymised, regarding both company name and individuals. This strategy has, in retrospect, proven successful on more than one occasion.

2.3.2 Stage two: Data analysis

To analyse the identified TCSs and consignments, Object-Oriented Analysis (OOA) is used. Object-orientated modelling languages such as UML provide a number of different visualisation tools together with a well-documented notation system. More precisely, the methods and tools in Object-Oriented Analysis and Design are well suited for the data present in this thesis.

Object-Oriented Analysis (OOA) is well documented in the OO-literature. Booch provides a definition:

“Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary in the problem domain.”

(Booch, 1991, p. 37)

Again, the importance of language is stressed. Booch, above, states the importance of vocabulary when analysing a system. The same claim is made by Morgan as he stresses the importance of using good metaphors when studying a subject (Morgan, 1980). Brunsson states that science affects systems “by creating languages to describe and understand” (translated from Brunsson, 1982). In OOA, terms and names of entities are preserved in the various diagrams that are used to visualise the analysed system from multiple perspectives.

An object-oriented analysis of the empirical data from the case studies in this thesis will focus on what Booch calls the “requirements from the perspective of the classes and objects”. In concrete terms, this means a focus on what a TCS requires of the consignments and vice versa.
By using the methods and tools of OOA to analyse case data, several benefits can be observed:

- A cross-case analysis requires a standardised notation and classification. Object-orientation relies on such a notation and classification system.
- Complex systems can be visualised with object-oriented modelling, enabling qualitative analysis.
- A future design phase will already be in progress. Much of the decomposition and classification takes place in the analysis phase; the design process can draw upon this, thereby saving time and effort, as well as ensuring a relationship with the analysis data.
- Classes and objects are defined once, and can thereafter be reused when building other models. This reusability is one of the key benefits when using object-orientation to construct arbitrary systems.
- Both static and dynamic system behaviour can be modelled, using the same set of tools and notations.
- Anyone literate in object-oriented modelling can understand a model, even if it depicts a type of system not previously encountered. For instance, a computer programmer will most likely be able to understand the models of transportation systems in this thesis, even though they are quite complicated and information-dense.

The table below is an extension of Table 1 on page 8. The last column shows the OO-diagrams that can be used to visualise the complexity drivers:

<table>
<thead>
<tr>
<th>Complexity type</th>
<th>Measurement</th>
<th>Driver</th>
<th>OO-diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptive</td>
<td>Variety, connectivity, interdependence</td>
<td>Size of state space</td>
<td>Class diagram</td>
</tr>
<tr>
<td></td>
<td>Information needed to describe the system</td>
<td></td>
<td>Statechart</td>
</tr>
<tr>
<td>Computational</td>
<td>Uniqueness, knowledge, transformation size</td>
<td>Number and length of valid trajectories</td>
<td>Class diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Statechart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sequence diagram</td>
</tr>
<tr>
<td>Uncertainty-based</td>
<td>Entropy, information needed to resolve embedded uncertainty</td>
<td>Number of decisions that must be made during the transformation</td>
<td>Class diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Statechart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sequence diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use case diagram</td>
</tr>
</tbody>
</table>

Variety, connectivity, and interdependence are all part of the system structure. There is one main structure-centred diagram in OO – the class diagram. In this diagram, a class including its attributes and operations is depicted together with its relationship to the other classes in the
system. The statechart is a graphical representation of the states that an object can assume in the system and gives, together with the class diagram, a visualisation of the total state space.

Sequence diagrams are used in OO to describe processes and series of events. The sequence diagram uses classes from the class diagram to achieve the states described in the state chart. Together with the statechart, the sequence diagram depicts not only each valid trajectory, but also its length (measured in the number of operations needed to achieve the goal state).

The system entropy is a measure of the total indeterminateness. When a model of a system fails to predict a state, a trajectory, or an operation in a sequence, entropy increases. To visualise entropy with OO-diagrams therefore becomes the visualisation of differences between studied systems and the models that were used when controlling them. The number of decisions, which had to be made during the studied transformation process, is a measurement that depicts the entropy that had to be handled in order to complete the transformation. The class diagram can be used to visualise the need for better interfaces, either inbound or outbound, in the TCS. The statechart can depict any states that are wrongly defined or missing. The sequence diagram can indicate whether the process has to be redefined in order to handle the uncertainty.

Another diagram that can be used is the use case diagram. The use case diagram depicts a part of the transformation, i.e. a part of the sequence diagram. The use case diagram is the diagram that most resembles the traditional narrative case study. A use case is a requirements analysis where a system, either an existing system or a planned one, is analysed based on its domain processes (Larman, 1998, pp. 8-11). Use cases are used in OOA to find natural boundaries for the different processes that exist within a system.

When conducting an analysis, regardless of method used, it is important to define a rudimentary analysis process early in the planning stage. This process is the basis for the collection and subsequent presentation of the data. In Table 4 above, the requirements for the data gathering are presented. Once the data is in place, the analysis can begin.

The OOA process for this thesis consists of the following elements:

1. Construction of object-oriented model of the case
   a. Construct class diagrams
   b. Construct statecharts
   c. Construct use case diagrams
   d. Construct sequence diagrams

2. Address the issues in Table 4 above using the model.
   a. Use the class diagram to identify the attributes, operations, and relationships between the objects in the system
   b. Use the statecharts to identify states and trajectories
   c. Use the sequence diagrams together with statecharts and class diagrams to find the valid trajectories and their length
   d. Combine the use case diagrams with the previous diagrams to find the various interactions between the objects in the model. Together with the narrative description of each case, any uncertainties may be identified.

When using OO-diagrams as visualisation of complexity, the models constructed are based on data from the modelled system as well as the researcher’s preconceptions and knowledge. There is not one “true” OO-model for any system, but rather one for each researcher where choices of abstraction level and internal boundaries between objects are subject to the
modeller. What all OO-models have in common are the basic principles upon which they are constructed, and the language that is used to describe phenomena.

2.3.3 Stage three: Designing future systems

This stage in the research takes a normative approach. The analysis of the collected data is used to formulate design recommendations addressing research question Q3.

Closely related to OOA is Object-Oriented Design (OOD):

“Object-oriented design is a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design.”

(Booch, 1991, p. 37)

The relationship between OOD and OOA is often discussed. Booch claims that the product of an OOA can serve as a model for an OOD, and the product from the design can serve as an implementation blueprint for Object-Oriented Programming (OOP):

“Object-oriented programming is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes via inheritance relationships.”

(Booch, 1991, p. 36)

OOD differs from OOA mainly in the “design” part of the definition. OOA never claims to design a future system whereas this is just what OOD strives towards. Larman (1998) explains how analysis and design activities are part of a continuum. The process of OOA is not merely accounting, and may well contain creative elements normally found in OOD. The distinction, he means, lies in the intent of the researcher, be it investigation (OOA) or solution (OOD).

More analysis-oriented

- what
- requirements
- investigation of domain

More design-oriented

- how
- logical solution

Figure 10 Analysis and design activities exist on a continuum (redrawn from Larman, 1998, p. 15)

Research question Q3, “How should the interface between the consignment and its Transport Control System be designed in order to minimise the impact of complexity in the control of the transportation system?” is addressed through the framework of OOD.

An important synthesis of the data analysis consists of a class library, where entities that are found in all or several of the cases are combined and presented as generic classes. These classes can be reused, and even extended, when building new models. The data collection/data analysis therefore constitutes the foundation upon which the design recommendations will be constructed.
3. FRAME OF REFERENCE

As was stated earlier in Chapter 1.2.1, this thesis is based upon three theoretical areas: Transportation science, Systems science, and Object-orientation.

The theories of transportation help define the problem domain and where to draw the system boundaries. The area of risk analysis theory is used when defining Goods Requiring Special Attention, itself a building block for this thesis. Systems theory facilitates the use of several well-defined models, such as the Black Box. Object-orientation (OO) is a powerful modelling philosophy that enables several views of a system, describing class- and object hierarchies as well as specific cases and generic rules. OO is applied systems science with close ties to the concepts of cybernetics, although with a sophisticated visualisation scheme not found in cybernetics.

3.1 Transportation

Transportation as a theoretical area can be approached in several ways, depending on viewpoint and the nature of the inquiry.

Encyclopædia Britannica defines transportation as:

“the movement of goods and persons from place to place and the various means by which such movement is accomplished.”

(Encyclopædia Britannica, 2007)

The term transportation consists of two parts. The movement itself – transport, and the means by which this movement is accomplished – traffic. Transportation is therefore the combination of transport and traffic. Using the nomenclature of cybernetics, a transport is a state transition where the end state differs from the initial state in geographical terms. The difference in states can of course be larger than just the transition in geography. In this thesis, the utility that a transport operation creates, i.e. the motivating mechanism behind the decision to transport, is not considered.

Sjöstedt identifies four disciplinary perspectives and describes them within the context of his model of goods, vehicles, facilities and infrastructure (Sjöstedt, 1997; Sjöstedt, 2005). The four structural components (the corners) in the model are connected through processes. Transport is the process that connects goods and vehicles/vessels, and traffic is the process between vehicles/vessels and infrastructure. Each corner represents a disciplinary perspective where transportation is regarded as the sum of transport and traffic with vehicles/vessels in the focus of attention:
Sjöstedt’s disciplinary perspectives are (Sjöstedt, 2005):

- Single mode transportation
- Physical planning
- Logistics
- Supply Chain Management (SCM)

Therefore, according to Sjöstedt, the difference between transportation and the other perspectives lies in the entity focus, be it goods, facilities, infrastructure, or vehicles/vessels. Logistics relates to transportation through a dialectic relationship, as is indicated in Figure 12 below.
In the figure above, the logistics system consists of three structural elements: Products, Locations and Facilities. The transportation system consists of Vehicles/Vessels, Freight and Ways & terminals. Product and Freight both represent the Goods from Figure 11 above, but from different perspectives. The same goes for Location and Ways & terminals as two views of the Infrastructure element. Logistics and Transportation meet along this diagonal in the figure through their respective processes Forwarding and Movement.

### 3.1.1 Logistics

Logistics can be defined in a number of different ways. Encyclopædia Britannica adopts a fairly straightforward definition:

“the organised movement of materials and, sometimes, people”

(Encyclopædia Britannica, 2007)

This definition can be compared to the definition from Council of Supply Chain Management Professionals (CSCMP):

“The process of planning, implementing, and controlling procedures for the efficient and effective transportation and storage of goods including services, and related information from the point of origin to the point of consumption for the purpose of conforming to customer requirements.”

(CSCMP, 2006)

In the Encyclopædia Britannica, as well as the CSCMP definition, financial goals are non-existent. Logistics is viewed upon as a cost driver that should be kept “efficient and effective”. Christopher, on the other hand, focuses on the process of profitability through cost-effective order fulfilment.

“Logistics is the process of strategically managing the procurement, movement and storage of materials, parts and finished inventory (and the related information flows) through the organization and its marketing channels in such a way that current and future profitability are maximized through the cost-effective fulfilment of orders.”

(Christopher, 1998, p. 4)
Where this thesis is concerned, the definition by Christopher together with Figure 12 clearly defines the term logistics and also sets it apart from Transportation, which is the main area of focus in this thesis (as previously stated in Chapter 1.2.1).

### 3.1.2 Models in transportation and logistics

Transportation and logistics systems can be studied on several levels, and from several vantage points. Three distinct model types are identified:

- **Relational models.** These describe relations between components in systems on a structural level.
- **Chronological models.** These models describe events and processes that can be put into a sequence.
- **Snapshot models.** These models describe a system in suspended animation, i.e. at a certain point in time.

The relational models are representations of how entities such as concepts, physical objects, or functions relate to each other. These models can be operationalised into the two other types, where processes and functions can be modelled, see Figure 13 below:

![Figure 13 Three types of models found in transportation science (Arnäs, 1999a).](image)

The model types can be seen as interpretation filters that can be used to study the same system from different aspects.

**Relational models**

The relational models are mainly conceptual, and are often used to order and label knowledge in relation to other knowledge.

In OECD (1992), transport operations are described in a hierarchical structure as situated between the material flow and the infrastructure. The material flow meets transport operations on the transport market where demand (material flow) is matched to supply (load unit flow).
On the traffic market, the transport operation layer presents a demand (vehicle flow) that is matched by the infrastructure supply (capacity). This model can be related to Sjöstedt (2005) as can be seen in Figure 11 where transport connects goods and vehicles/vessels and traffic connects vehicles/vessels with infrastructure.

In another model from OECD, the various roles in a logistics system are mapped and their relations are described. The model is used to describe how various categories of organisations are involved in the logistics system (OECD, 1996).

Yet another model describes critical systems architectures and the effects of redundancies by distinguishing between two types of networks. In *series networks*, all components must operate correctly in order to achieve the system goal. In *parallel networks*, all components must fail in order to cause total system failure (Blanchard, 1998).

Powell (2001) describes, in a relational model, the transport system from the viewpoint of policy makers and with a clear infrastructural perspective.

**Chronological models**

When adding time as a component in modelling, processes and series of events can be studied. Models handling routing problems are examples of chronological models. Such models can for instance be found in Ortúzar and Willumsen (2001), where a number of mathematical models on for instance vehicle routing can be found.

Another perspective is taken by Källström, who models transportation flows where responsibilities are related to the flow of goods between actors (Källström *et al.*, 2000).

**Snapshot models**

All snapshot models have one thing in common. They describe dynamic systems in suspended animation, enabling analysis of not only relations between elements, which normally are dynamic (“moving”), but also measurement of change (“velocity”) or, if more than one snapshot is taken, speed of change (“acceleration”). The snapshot models are often used when systems are too complex to study in real time, giving a researcher the opportunity to describe “samples” that, when put together, form a chronological approximation of a phenomenon.

One well-known example of a snapshot model is the EOQ-model (e.g. Lambert *et al.*, 1998). This model describes the mechanism between ordering costs and inventory, carrying cost for a certain period of time. The model calculates the total cost *at a certain time* based on a number of decisions regarding order quantity, frequency etc.

Another example is the model of “Logistical nodes” that can be modelled as a sequence of translations (geographical rearranging) and transformations (changing appearance). In each of these nodes, handling units are used to perform the translations/transformations (Vermunt and Ruijgrok, 1993).

**3.1.3 Models used in this thesis**

Each of the entities (corner components) in Sjöstedt’s model represents different levels of abstraction. The associated relationships serve to connect the components not only in the physical sense, but also in time. Franzén uses a logarithmic time-axis to explain the different time horizons in public transport (Franzén, 1999). His model can be modified to explain the

---

6 EOQ = Economic Order Quantity. The EOQ-formula was first introduced in 1915 by F.W. Harris
temporal connections that are made in Sjöstedt’s model as well as to position this thesis within that framework (as suggested in Sjöstedt, 1997 p. 28):

Figure 14 This study positioned within two frameworks (modified from Franzén, 1999; Sjöstedt, 1997).

Studying the time scale in the figure above gives that the interval treated in this thesis lies between one minute up to a couple of days. In practical terms this has implicated the data collection process, so that information outside this “time window” has been excluded from the case studies. By taking such a stance, the abstraction levels of both vehicle/vessel as well as infrastructure fall outside the main scope of the thesis.

Transport Control System, TCS

The term Transport Control System – TCS – is defined on page 5 as:

A TCS is a system that controls the trajectory of a transportation process.

The TCS is here seen as a regulator that manipulates the inbound interface of the transportation system in order to produce the desired trajectory.
The view of the TCS as a controller is reminiscent of the term Transport Co-ordinator, as described by Ramstedt and Woxenius (2006). They divide the actors in a transport chain according to the five abstract terms Source, Sink, Management, Link operator, and Node operator:

*Table 6 Categories of transport chain actors (modified from Ramstedt and Woxenius, 2006). Additions in bold type.*

<table>
<thead>
<tr>
<th>Abstract terms</th>
<th>Generic actor names</th>
<th>Roles</th>
<th>Practically used actor names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Consignor</td>
<td>Send goods</td>
<td>(Product) Supplier</td>
</tr>
<tr>
<td>Sink</td>
<td>Consignee</td>
<td>Receive goods</td>
<td>(Product) Customer</td>
</tr>
<tr>
<td>Management</td>
<td>Transport Co-ordinator</td>
<td>Co-ordinate transport services</td>
<td>Forwarder, Third-party logistics provider, Agent</td>
</tr>
<tr>
<td>TCS</td>
<td>Control transportation process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link operator</td>
<td>Transport operator</td>
<td>Move goods</td>
<td>Road haulier, Rail operator, Shipping line, Airline</td>
</tr>
<tr>
<td>Node operator</td>
<td>Terminal operator</td>
<td>Tranship, consolidate, or deconsolidate goods</td>
<td>Port, Airport, Intermodal terminal operator, Consolidation terminal operator</td>
</tr>
</tbody>
</table>

A single company can assume one or more of these roles, e.g. large industrial companies sometimes have their own forwarding departments and so on. The only one of these roles that has a view of the entire transport chain is the TCS. The TCS is, potentially, also the one that will be most affected by differences in goods type.

Networks and trajectories

Depending on the focus and hierarchical level of an observer, a transportation system can assume the form of either a chain or a network. For the single consignment that has just arrived at the consignee, the system is a chain of nodes and links that connects the consignee with the consignor, in this thesis called a *trajectory* (see page 4 for definition).

When taking the perspective of the single consignment, there is no need to look at other nodes and links outside the single trajectory. The TCS which is responsible for the transportation, on the other hand, will regard the system as a network of nodes and links, where each node provides options to choose among several alternative links. The network consists of nodes and links and can represent networked relationships on several levels, such as information, infrastructure, and resources.

The network represents the large variety of choices that are available when sending goods. For each node, there is often more than one possible link that can be chosen for the continuation of the transport. A network model is structural and does not explicitly include the process of choosing links. However, once the links are chosen, the path through the network takes on the form of a trajectory.

Network models can be hierarchical in nature. Each node or link can consist of several other nodes and links on a lower level.
Figure 15 A network model can be hierarchical. Each node or link can contain other nodes and links. In this figure, a link in the displayed trajectory is expanded into three new links and two nodes.

This means that the resolution of a trajectory can be increased or decreased depending on what is studied. As seen in Figure 14, the time horizon defines the level of abstraction and vice versa. The same reasoning is applicable to the choice of resolution when studying a single trajectory. The lower the abstraction level, the higher the resolution needed.

Terminals

Terminal facilities exist in many transportation systems. A terminal facility is a physical structure (often a building) that accommodates the needs of different vehicles in terms of loading equipment, storage areas, information handling etc. In this document, the term terminal is used in its broadest sense, as defined by Hultén (1997). Hultén defines the terminal function as bridging gaps between processes:

Figure 16 The terminal function is to bridge the gaps in frequency, capacity and time (redrawn from Hultén, 1997).

By using this definition, any node where goods can be shifted between links is a terminal. An example of a terminal is a port. Containers arrive from the hinterland by lorry at the port with the frequency of 20 per hour during office hours. A ship with the capacity of 800 containers leaves once every evening. The port area is used to store the containers until they can be loaded onto the ship, thereby bridging the gaps of frequency (20 per hour vs. once per day), capacity (1 vs. 800) and time (containers that arrive in the morning will be transported with the ship in the evening). A well functioning terminal facilitates high resource utilisation on both sides (i.e. that the actual load matches C1 and C2 respectively).
In a sense, all nodes in a network can be considered terminals. A node fulfils each of the three criteria of bridging the gaps in frequency, capacity, and time. Therefore, it can be stated that a node fulfils the *terminal function* without the necessity of being a *terminal facility*.

In road-based transport, the consignment size is often large enough to reduce the need for terminal handling and consolidation to maintain a high load factor. A road transport of this type is called FTL – Full Truck Load.

> "FTL: A term which defines a shipment which occupies at least one complete truck trailer, or allows for no other shippers goods to be carried at the same time."

(CSCMP, 2006)

When terminal facilities are involved, the transport is called LTL – Less than Truck Load.

> "LTL Carrier: Trucking companies that consolidate and transport smaller (less than truckload) shipments of freight by utilizing a network of terminals and relay points."

(CSCMP, 2006)

Depending on the terminal layout, a trajectory may take a distinctly different form from one studied case to another.

Woxenius (2007) identifies six alternative ways to organise terminal facilities in a network, as illustrated in Figure 17 below.

![Figure 17 Six options for transport from origin (O) to destination (D) in a network consisting of ten terminals (Woxenius, 2007).](image)

A *direct link* transport is not influenced by any of the other terminal facilities. The consignment is transported directly from the consignor to the consignee. In the *corridor* layout, there is a high-density flow (called *artery*) which is served by smaller flows (capillary). The *hub-and-spoke* layout consists of a central terminal facility acting as a sorting station. Every consignment is transported to the hub, sorted and then transported to the destination together with other consignments bound for the same terminal facility. Terminals are either *hub* or *spoke* terminals. The *connected hubs* layout is hierarchic and described as a direct link with regional consolidation. *Static routes* consist of a pre-defined series of links, forming chains or loops that are visited by the transporter in sequence. Each terminal facility exchanges goods with the transporter. The most flexible layout is *dynamic routes*, where each transport is individually planned. Woxenius states that the layout of complex transportation networks often are combinations of the types listed above, and also that various actors can classify the same network differently depending on their viewpoint (Woxenius, 2007).
Therefore, a haulier may classify a system as direct link whereas the forwarder that employs him envisions it as connected hubs.

3.1.4 Defining heterogeneous goods

The term heterogeneous goods is central to this thesis. This section contains two parts; first, the previous work regarding this term is accounted for; second, a new definition is presented and put into the context of the transportation process described in Chapter 1.2.2 on page 4.

Previous work

The term Goods Requiring Special Attention (GRSA) was introduced in (Arnäs, 1999a). The reasons for special attention requirements included a risk-hazard assumption by the system where probabilities and consequences of different disturbances were estimated.

A disturbance is an unplanned event causing negative consequences, and thus, the potential for a disturbance involves some amount of uncertainty or probability. The consequences of a disturbance can manifest themselves in a number of ways, for instance:

- Accident causing personal injuries, or death
- Litigation caused by missing goods
- Dissatisfied customer due to delays in transportation
- Misplacement in warehouse due to faulty labelling of goods

The disturbance potential is a function derived from risk analysis theory where probability and consequence of different events are plotted in the same diagram. The same methodology can be applied here if the probability and consequence of a disturbance is used. The diagram was assumed to take on the characteristic form of an FN-curve as shown in Figure 18 below. F stands for Frequency and N for Number of fatalities (in risk analysis theory).
In the FN-diagram a number of scenarios are plotted, each scenario is defined by likelihood to occur and a consequence if it occurs. Instead of using “scenario” as a parameter in each data triplet, the term disturbance is used:

\[ d_i: \text{ a disturbance identification or description (in risk analysis this parameter is called } s_i \text{ for “scenario”)} \]

\[ p_i: \text{ the likelihood of that disturbance} \]

\[ q_i: \text{ the consequence of the disturbance (cost)} \]

The curve represents the defined situation’s disturbance potential instead of risk.

\[ D_p = (q_i, P_i) \]

Different goods types require different amount of attention from the system. Goods Requiring Normal Attention (GRNA) are the goods that the system was designed for (homogeneous goods). Goods Requiring Special Attention (GRSA) are all those types that the system is not designed for but is able to handle with special attention given (heterogeneous goods).

There are two major types of attention requirements; external and internal. External attention requirements are formulated by actors outside the system and are, on the operative level, non-negotiable. Internal attention requirements are formulated within the system and can be both factual and perceived, as shown in Figure 19. The factual requirements are formulated in operation manuals, company policies, work orders etc. The perceived attention requirements are based on estimations done by one or more persons in the system, and they serve as an “uncertainty buffer”, where concepts like “worst-case scenarios” or “expect the unexpected” are common.
The system’s goal is to estimate the factual disturbance potential as closely as possible and then compensate for any increase by adapting existing operations or by creating new ones.
A new definition
Lambert et al. describe five factors that influence the price/cost of transportation for a product (Lambert et al., 1998):

- **Density, D**
  - The weight-to-volume ratio influences the transportation cost, mainly because low density products, like tissue paper, never utilise the full weight of the load carrier.

- **Stowability, \(^7\) W**
  - The degree to which a product can fill the available space in a load carrier. The concept of *cube utilisation* is used, which means that the ideal situation occurs when the product completely fills an imaginary cube.

- **Ease or difficulty of handling, H**
  - Products with poorly designed interfaces or with non-uniform physical properties are more costly to transport.

- **Liability, L**
  - When the value-to-weight ratio is high, the product often carries a higher transportation cost due to increased liability of the transporter.

- **Market, M**
  - In addition to the product-related factors above, there are several market-related factors that influence the cost/price of a transport:
    - Competition
    - Location of markets
    - Regulations
    - Traffic balance
    - Seasonality changes
    - Internationality

The transportation cost, \(C_T\) is therefore a function of the four variables, D, W, H, and L together with the market-related factors, M.

*Equation 2 The cost of a transport is a function of the four parameters Density, Stowability, Ease of Handling, Liability, and Market*

\[
C_T = f(D, W, H, L) + f(M)
\]

It is assumed that \(f(M)\) is constant in relation to the variables studied in this thesis and will therefore be disregarded in future equations.

---

\(^7\) The letter W is used to denote Stowability. The letter S is used to denote State elsewhere in this thesis.
For a transportation system, the parameters, D, W, H, and L are often regulated in the pricing strategy, for example (DHL, 2006; DFDS Transport, 2006):

- **D**: When the density is lower than 333 kg/m$^3$, the price is calculated based on the assumption that the actual density is 333 kg/m$^3$ leading to a higher calculated weight, in turn leading to a higher cost.
- **W**: The length density is calculated to 1850 kg/metre of the load unit. This means that goods that cannot be consolidated (for instance no other goods can be stowed on top) are calculated to a weight that is higher, leading to a higher cost.
- **H**: Consignments exceeding the following measurements cost extra due to handling difficulties: height > 2,5 m, length > 6 m, width > 2.4 m.
- **L**: Temperature-sensitive, dangerous, or theft-prone goods are subject to prior notification to the transporter and connected to a cost increase.

This means that in most transportation systems, there are limits regulating the product-related parameters in the cost function:

\[
C_T = f(D, W, H, L)
\left\{ \begin{array}{c}
D_{\text{min}} \leq D \leq D_{\text{max}} \\
W_{\text{min}} \leq W \leq W_{\text{max}} \\
H_{\text{min}} \leq H \leq H_{\text{max}} \\
L_{\text{min}} \leq L \leq L_{\text{max}}
\end{array} \right.
\]

Goods with all the parameters D, W, H, and L inside their accepted intervals are called **homogeneous goods**:

**Homogeneous goods are defined as goods where none of the parameters Density (D), Stowability (W), Ease of handling (H), or Liability (L) are outside their accepted ranges.**

When one or more parameters are found outside the accepted interval, a new cost function is needed:

\[
C'_T = f(D', W', H', L')
\]

**Heterogeneous goods are defined as goods where one or more of the parameters Density (D), Stowability (W), Ease of handling (H), or Liability (L) are outside their accepted ranges.**

To emphasise that the goods are heterogeneous, $C'_T$ is used to denote their cost function.

Relating the cost function to the transportation process function from Chapter 1.2.2 is not straightforward (see Figure 2 on page 4). The parameters D, W, H, and L are not directly represented in the state space $S$ of the transportation process model. Both models depict the same process, but with different motives. The transportation process function describes what happens, and the cost function describes the cost of what happens. In essence, this means that the transportation process, $T$, generates the cost $C_T$. 

48
The cost function, $C_T$, as shown in Equation 3 above relates to the transformation function as follows:

*Equation 5* The Density ($D$), Liability ($L$), Stowability ($W$), and ease of Handling ($H$), are functions of the states as well as of the resources.

\[
T = f(\tilde{S}_0, \tilde{S}_1, \tilde{R}) \quad \begin{cases} 
D = f(\tilde{S}_0) \\
W = f(\tilde{S}_0, \tilde{R}) \\
H = f(\tilde{S}_1 - \tilde{S}_0, \tilde{R}) \\
L = f(\tilde{S}_1 - \tilde{S}_0)
\end{cases}
\]

The Density, $D$, is a function of the initial State, $\tilde{S}_0$. $D$ consists of physical parameters inherent in the consignment and present in the initial state.

The Stowability, $W$, is a function of the initial state and the resources used, $\tilde{R}$. Stowability does not only represent physical dimensions but also the ability to fit into the chosen load unit, which causes the dependence on $\tilde{R}$.

The Ease of Handling, $H$, depends on the trajectory, $\tilde{S}_1 - \tilde{S}_0$, and the resources used. Since multiple trajectories often are possible, the ease of handling depends on the trajectory chosen, as well as on the resources available for it.

The Liability, $L$, is a function of the trajectory. The Liability changes with the choice of trajectory, as each alternative trajectory carries its own state transitions that may differ from other trajectories.
Added value and transportation cost

As shown in Arnäs (1999a), there are effects on transportation systems caused by discrepancies in goods types. The goal was to establish relations between differences in goods type and attention requirements. This was accomplished on a conceptual level, ending in the figure below. \( C_T \) and \( C'_T \) stand, as shown above, for the accumulated cost for the state change from \( S_0 \) to \( S_1 \) for homogeneous and heterogeneous goods respectively.

![Figure 20 Cost increase of heterogeneous goods according to Arnäs (1999a).](image)

Arnäs (1999a) tried to measure the actual cost of heterogeneous goods, which turned out to be difficult. To measure the cost increase for a single consignment in absolute terms is nearly impossible and would not be very productive since the data requirements for such a calculation would be substantial. The question may be wrongly phrased, however. Instead of focusing on immeasurable costs, the focus could be on the difference in state transitions that can be observed by studying goods with different properties. It is the state transitions that generate cost and the goods that require the state transitions.

A consignment is traditionally viewed as a passive entity, allowing it to be handled rather than to perform the handling. Thus, being passive by nature, the consignment does not perform any cost-driving activities. It can, however, communicate a need and even facilitate its fulfilment. The operations that are required by the consignment to fulfil its needs are performed by resources contained in the transportation system. These resources on the other hand generate cost. This means that the cost attributed to a consignment is derived from the state transitions required.

![Figure 21 State transitions are causing cost, not the goods.](image)
If, for instance, a consignment is due at noon tomorrow, the system needs to perform state transitions that ensure this. There are often several alternative ways to choose what individual transitions to perform, and as long as the consignment arrives in due time, the choice is up to the control system. Therefore, the cost is not directly linked to the goods, but rather to the choices that the control system makes in order to accommodate the requirements for the goods.

3.2 Systems science

This chapter will provide a brief insight into what is known as systems science, and more specifically, cybernetics.

General Systems Theory was first introduced in a seminar 1937 by the biologist Ludwig von Bertalanffy (Bertalanffy, 1973).

“General Systems Theory is a name which has come into use to describe a level of theoretical model-building which lies somewhere between the highly generalized constructions of pure mathematics and the specific theories of the specialized disciplines”

(Boulding, 1956)

The chapter will begin with a discussion of what a system is and how it can be defined. There are several “schools of thought” that are slightly different in this. Following the first part are three sections depicting the problems of describing, designing and controlling systems respectively (problems to which there are several approaches). An approach for this thesis is defined.

One important work is that of Churchman in *The Systems Approach* (Churchman, 1981). The entire book is in the format of a debate between advocates of four different system approaches, namely:

1. The advocates of efficiency
   - Identify trouble spots in a system and remove inefficiency
2. The advocates of science
   - Build an objective model of the system that describes how it works
3. The humanists
   - Systems are people, and the fundamental approach to systems consists of first looking at the human values
4. The anti-planners
   - The correct “approach” (*Churchman’s quotes*) to systems is to live within them

Churchman concludes (Churchman, 1981, p. 230) that “there is bound to be deception to any approach to the system”. He also presents what he refers to as his own bias: “The systems approach is not a bad idea” (Churchman, 1981,p. 232).

3.2.1 What is a system?

If Churchman is to be believed, the systems approach is dangerous ground for a scientific traveller. He implies that there are no “truths” about any system, and that the approach decides not only the results of a study, but also the set of tools available to the researcher. He
(deliberately?) avoids explicit definition of the word ‘system’. Although there are a few occurrences of definitions in the book, they are within the context of one of the four advocates mentioned above. Common for them is that a system is “a set of parts coordinated to accomplish a set of goals”. (Churchman, 1981 pp. 11, 29)

Even if there are many types of advocates, as Churchman phrases it, there is a consensus in the systems research area regarding some of the characteristics of a system:

- A system is composed of more than one (elementary) part
  - There are always what can be called elementary parts, building blocks, which are not destroyed or divided into other parts within the time horizon within which the system is studied
  - For something to be called a system, it must contain at least one set of such parts. The set must contain at least two parts.
- A system differs from a set when it displays emergent properties
  - When a set displays some function or property that is related to the whole, this property is called emergent property. The set is then a system.

One common denominator in many definitions of systems is that the boundaries are as important to describe as the system that is contained within them. Ackoff proposes a system as an “indivisible whole” that loses its essential properties when taken apart:

“A system is a set of interrelated elements of any kind; for example, concepts (as in the number system), objects (as in a telephone system or human body), or people (as in a society). The set of elements has the following three properties.

1. The properties or behavior of each part of the set has an effect on the properties or behavior of the set as a whole. For example, every organ in an animal’s body affects the performance of the body.

2. The properties and behavior of each part and the way they affect the whole depend on the properties and behavior of at least one other part of the set. Therefore, no part has an independent effect on the whole. For example, the effect that the heart has on the body depends on the behavior of the lungs.

3. Every possible subgroup of elements in the set has the first two properties. Each has an effect, and none can have an independent effect, on the whole. Therefore, the elements cannot be organized into independent subgroups. For example, all the subsystems in an animal’s body – such as the nervous, respiratory, digestive, and motor subsystems – interact, and each affect the performance on the whole.”

(Ackoff, 1973, pp. 327-328)

The predominant way to study a phenomenon has for long been the use of reductionism. In short, this means that a phenomenon can be fully understood by understanding its parts. If, after gaining understanding of every part, the phenomenon still cannot be understood, there is insufficient data regarding one or more of the parts. In systems science, where a system is said to contain interacting subsystems, it is not possible to use reduction. The strict hierarchical relationships that must apply for reductionism to work do not exist. Ashby attacks reductionism by the following example:

“Now combination by simple addition is the very next thing to no combination at all. Thus one penny combines with one penny to give just two, precisely because pennies do not in fact interact to any appreciable extent. Contrast this merely nominal combination with what happens when, say, acid is brought together with
alkali, or rabbit is brought together with rabbit. Here is the real interaction, and the outcome cannot be represented as a simple sum.”

(Ashby, 1958, p. 250)

Checkland (1976) and Checkland and Scholes (1990) prefer to identify systems as pertaining emergent properties, thereby negating a reductionist approach to them.

“The systems movement is identified by a conscious use of the concept "system" and by the holistic thinking which it implies. The movement’s holism is best understood with reference to its opposite: reductionism. The reductionist is committed to explanation in terms of the smallest number of the most fundamental entities, which in science means those of physics. /.../ But even in physics and physical chemistry there are phenomena – such as those connected with heat flow – which have no meaning at all in terms of individual atoms and molecules but which are repeatably measurable and which lead to theory able to explain the observations. Such emergent properties are characteristic of a given level of complexity.”

(Checkland, 1976, p. 264)

“[the most basic core idea of systems thinking is] that a complex whole may have properties which refer to the whole and are meaningless in terms of the parts which make up the whole. These are the so called ‘emergent properties’.”

(Checkland and Scholes, 1990, pp. 18-19)

Checkland (and Scholes) and Ashby argue that systems science is a more “complete” way of structuring the world, and that reductionism is insufficient. Gougen and Varela take a more humble approach when addressing the notion of reductionism versus holism and conclude that they complement each other:

“One thing to notice is, that in the hierarchy of levels, "emergent" or "immanent" properties appear at some levels. {wholeness} can be measured by the difficulty of reduction. /.../ It is not that one has to have a holistic view as opposed to a reductionist view, or vice versa, but rather that the two views of systems are complementary.”

(Gougen and Varela, 1979, p. 299)

Checkland and Scholes also mean that a system cannot be defined from within. The very concept of a “whole” negates the possibility of anyone within this whole having sufficient knowledge to describe it.

“Unfortunately he {von Bertalanffy} made a bad mistake in using the word ‘system’ for the name of the abstract notion of a whole he was developing. /.../ {von Bertalanffy} clearly regards ‘system’ as an abstract concept, but unfortunately he immediately starts using the word as a label for parts in the world. /.../ Choosing to think about the world as if it were a system can be helpful. But this is a very different stance from arguing that the world is a system, a position which pretends to knowledge no human being can have.”

(Checkland and Scholes, 1990, p. 22)

Beer takes a pragmatic approach:

“The system we choose to define is a system because it contains interrelated parts, and is in some sense a complete whole in itself.”
Beer categorises systems in two dimensions; the degree of complexity and whether or not the system is deterministic. This is an abstraction that needs reflection. Consider the table below:

Table 7 Categorisation of systems according to Beer (redrawn from Beer, 1959, p. 18)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Simple</th>
<th>Complex</th>
<th>Exceedingly complex</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deterministic</strong></td>
<td>Window catch</td>
<td>Electronic digital computer</td>
<td>Empty</td>
</tr>
<tr>
<td></td>
<td>Billiards</td>
<td>Planetary system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Machine-shop layout</td>
<td>Automation</td>
<td></td>
</tr>
<tr>
<td><strong>Probabilistic</strong></td>
<td>Penny tossing</td>
<td>Stockholding</td>
<td>The economy</td>
</tr>
<tr>
<td></td>
<td>Jellyfish movements</td>
<td>Conditioned reflexes</td>
<td>The brain</td>
</tr>
<tr>
<td></td>
<td>Statistical quality</td>
<td>Industrial profitability</td>
<td>The company</td>
</tr>
<tr>
<td>control</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As described in Chapter 1.2.7, the term *complexity* and its application on transportation and logistics has recently been extensively covered by several authors (Nilsson, 2006; Nilsson, 2005; Nilsson, 2003; Waidringer, 2001; Franzén, 1999; Hultén, 1997). Waidringer defines complexity as the difficulty in modelling a system. Beer defines complexity as absence of determinism. Recent findings indicate, however, that a highly complex system can emerge from extremely simple and deterministic rules (Wolfram, 2002). Wolfram discusses the uses of *Cellular Automata* as a model of not only an arbitrary system, but also the Universe and Time itself. If he is correct in his bold statements, it means that the table above needs correction. One fact remains, however: complex systems, however they are put together, will not be adequately described using reductionist methods.
In Table 8 below – the same table as Table 1 on page 8 – three types of complexity are identified. Each type has a corresponding measurement and a driver:

Table 8 Three complexity types, their measurements, and corresponding drivers.

<table>
<thead>
<tr>
<th>Complexity type</th>
<th>Measurement</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptive</td>
<td>Variety, connectivity, interdependence</td>
<td>Size of state space</td>
</tr>
<tr>
<td></td>
<td>Information needed to describe the system</td>
<td></td>
</tr>
<tr>
<td>Computational</td>
<td>Uniqueness, knowledge, transformation size</td>
<td>Number and length of valid trajectories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncertainty-based</td>
<td>Entropy, information needed to resolve embedded uncertainty</td>
<td>Number of decisions that must be made during the transformation</td>
</tr>
</tbody>
</table>

Each of these three types is described further in Chapter 1.2.7 on page 7.

### 3.3 To describe and design a system

One of the most powerful and predominant tools in science is the use of models. The building of a model of a system is an active intellectual process. The scientist applies knowledge and data together with logic into a creative abstraction of reality. This abstraction is a model. Wolfram extends a warning for overconfidence in models. According to him, they are products of their creator and are therefore susceptible to subjective idealization.

> “Any model is ultimately an idealization in which only certain aspects of a system are captured, and others are ignored. /.../ One might perhaps think that in the end one could always tell whether a model was correct by explicitly looking at sufficiently low-level underlying elements in a system and comparing them with elements in the model. But one must realize that a model is only ever supposed to provide an abstract representation system – and there is nothing to say that the various elements in this representation need have any direct correspondence with the elements of the system itself.”

(Wolfram, 2002, p. 364-365)

The challenge of a systems scientist is to build good models of the systems that are studied.

> “/.../ the point of making models is to be able to bring a measure of order to our experiences and observations, as well as to make specific predictions about certain aspects of our experienced world.”

(Casti, 1989 p. 2)

According to Casti, above, making a model roughly means “to order” the environment in such a way as to understand or even predict it. Modelling is a tool that humans use to rationalise their surroundings. There is a twofold purpose of modelling. Klir (1988) explains the duality of modelling as being used for both inquiry as well as design:
“I have always considered modelling as the central problem area in systems science. In my opinion, it encompasses both systems inquiry and systems design. In both cases, we attempt to construct systems (at appropriate epistemological levels) that are adequate models of something of our interest: either some aspects of nature or some aspects of desirable man-made objects. In systems inquiry, we construct models for the purpose of understanding the phenomenon of inquiry, making adequate predictions or retractions, controlling the phenomenon in a desirable way, and making appropriate decisions; in systems design, we construct models for the purpose of prescribing operations by which a desirable artificial object can be made to satisfy objective criteria within given constraints.”

(Klir, 1988, p. 349)

It is therefore of importance to realise that in systems science, modelling can be either an inquisitive or a creative process, depending on the purpose of the activity. In retrospect, it can be difficult to deduce whether a model is the result of an inquiry or of a creative process without an explanation from the author. In this thesis, both approaches are used.

Rosen describes the work of Ashby in his lucid and honest review from 1985 (Rosen, 1985). According to Rosen, Ashby takes another approach and proposes that a system is an abstract object, consisting of a finite number of variables associated with a machine. According to his definition, a system is nothing more than a model of a reality too complex to fathom (by Ashby named machine). This view is somewhat opposite to the holistic statement from Ackoff on page 52. It is not justified, however, to label Ashby a reductionist as he clearly states that the whole indeed is a better place to start the study and then to diminish the scope until a more suitable set of objects to study is found (see also the quote from Ashby on page 52):

“Instead of studying first one system, then a second, then a third, and so on, it goes to the other extreme, considers a set of "all conceivable systems" and then reduces the set to a more reasonable size.”

(Ashby, 1958, p. 250)

Rosen suggests that any system can be described in two ways: internal and external. Internal description means that a snapshot is made of the system at a moment in time and its state variables are noted. The external description is of the dynamics of the system as “seen” from the outside. Rosen also seeks a way to combine these two description approaches:

“In systems for which the passive, autonomous aspect is paramount, a kind of system description is appropriate which we shall call internal description. Typically such a description begins with a characterization of what a system is like at an instant of time; such a characterization is said to define a state of a system. The totality of all the possible states of a system, meaning the totality of different aspects the system can assume for us at an instant of time, forms a set called the state space of the system. /.../ The fundamental problem of system description is to determine how the initial states change with time under the influence of the forces acting on the system. /.../ The temporal evolution of the system thus takes the form of a curve, or trajectory, in the state space.

The other kind of description is called an external description, sometimes graphically called a black-box description. In this situation we make no attempt to identify a set of state variables for the system. Rather we have at our disposal a family of perturbations which we can apply to the system, variously called system forcings or inputs, and one or more observables which we use to index the effect
of applying a particular forcing or input to the system. Such system observables are called system outputs or responses. /…/ 

It is one of the goals of science to be able to match up the two kinds of system description we have described. The external description is a functional one; it tells us what the system does, but not in general how it does it. The internal description, on the other hand, is a structural one; it tells us how the system does what it does, but in itself contains no functional content.”

(Rosen, 1972, p. 610)

While internal descriptions consist of a series of snapshots, the external ones are more concerned with cause and effect. Consider the figure below containing a system with a state space of 3. As can be seen when looking inside the system boundary, state 3 can only be reached from state 1. On the outside, however, the system consists of an interface with 3 inputs and one output.

![System Boundary Diagram](image)

Figure 22 In this system, there is no way to go from state 2 to state 3 without first passing state 1. This is directly observable if the system boundary is open. If the boundary is closed, the only way to interpret the system is via the input-output relation.

The differences between the internal and external descriptions are not merely about point of view. The two approaches are fundamentally different in that they regard the studied system as either affected or “affectable”. This distinction depends on the abstraction level of the observer. An internal description of a system can contain elements that are externally described, e.g. when looking inside the system boundary other systems may be found that are only visible externally. This kind of nested approach is adopted throughout systems science.

Buede describes a system using the definition similar to that of Churchman:

“[A system is a] set of components (sub-systems, segments) acting together to achieve a set of common objectives via the accomplishment of a set of tasks.”

(Buede, 1999, p. 38, 124)

According to Buede, a system is related to its surroundings on two levels (see Figure 23 below). The first level contains external systems. They are a “set of entities that interact with the system via the system’s external interfaces” (Buede, 1999, p. 38, 124). The system in focus is able to affect the systems on this level, as well as be affected by them. The second level is the context, by Churchman named the environment (Churchman, 1981, p. 36). This level affects the system, but is not affected by it.
When building a model of a complex system, there is clearly need for some kind of simplification (Weinberg, 1972). Ashby takes this reasoning a step further when he states that “… systems theory must become based on methods of simplification, and will be founded, essentially, on the science of simplification. /…/ The systems theorist of the future, I suggest, must be an expert in how to simplify.”

(Ashby, 1964, p. 510)

Casti distinguishes between “good” and “bad” models. A “good” model is signified by simplicity, a reasonable degree of data agreement, and explanatory power. By simplicity, Casti refers to Occam’s Razor and concludes that “/…/ when everything being equal (which it never is), take the simplest model that agrees with the observations.” His statement that nothing is ever equal leads to the second criterion, reasonable data agreement. This means that a “good” model agrees with a majority of the observations. The fact that there may be some observations that do not concur with the model is, according to Casti, one of the trademarks of a “good” model since these anomalies spur further development and in some cases even lead to scientific revolutions (he cites the famous discrepancies between Newtonian and quantum mechanics as an example of this). The third prerequisite for a “good” model is explanatory power. There should be “some interpretable connections between the entities of the formal system comprising the model and the physical entities characterizing the natural system under study” (Casti, 1989, pp. 469-470).

In other words, a good model should be simple, fairly accurate, and be able to act as a representation of the system it models in such a way that the model will react to input in the same way as the studied system. Conant and Ashby made the same claim when they stated that “every good regulator of a system must be a model of that system” (Conant and Ashby, 1970, p. 511). There is a trade-off in this list of criteria, however. The Law of Requisite Variety clearly demands that a regulator, i.e. a model of a system has to possess at least the same variety as the system it regulates (Ashby, 1956). Conant and Ashby even say that the

---

8 Occam’s Razor is a principle attributed to the 14th-century English logician and Franciscan friar William of Ockham. Translation of the original text reads: “Plurality should not be posited without necessity.” The principle gives precedence to simplicity; of two competing theories, the simplest explanation of an entity is to be preferred. Sources: Britannica Online and Wikipedia.
best regulator possesses exactly the same variety as the regulated system (Conant and Ashby, 1970). Therefore, to be able to model a system in such a way that the model gains high explanatory power, the model must contain a significant amount of the variety in relation to the studied system. At the same time, it should be kept simple. The key to the solution of this trade-off is the second criterion that states that the model does not have to be perfect, it just has to agree with most of the observable data on the studied system. Therefore, a simple model may well have significant explanatory power, but only for a subset of the total state space in a system.

3.3.1 Simplification strategies

How can a system model be simplified without losing explanatory power? According to Klir (1991) the four most important simplification strategies are (Klir, 1991 pp. 138-142):

- Exclude variables
- Partition states
- Break down into subsystems
- Organise subsystems hierarchically

Breaking a system down into subsystems is a well-known strategy similar to the idea of reductionism. However, as has been stated earlier, a system cannot be described in a satisfactory manner by adding the properties of each sub-system as is common practice in reductionism. This encapsulation is used to hide the internal function of a system. Function is encapsulated and replaced by an interface instead. This interface is the only way through which it is possible to interact with the encapsulated system.

The interface can be divided into two parts: inbound and outbound. The interfaces encapsulate the inner functions of the object, so that they for an observer actually become the object. There is no apparent difference between the interfaces and the object per se. The inbound interface consists of the system’s operations and the outbound of its attributes. The width of the interface depicts the number of parameters (i.e. the number of attributes/operations). It can vary from a select few parameters to several. The depth of the interface relates to the number of possible values each parameter can assume, which may vary from a binary switch to a continuum.

To achieve any sort of recognisable encapsulation, the interfaces must be at least finite in depth and width. Manageable encapsulations contain narrow and shallow interfaces. The famous 6th century BC Chinese general Sun-tzu wrote in his magnum opus ‘The Art of War’ the following on the importance of standardised interfaces within an army:
The Military Administration states: ‘Because they could not hear each other they made gongs and drums; because they could not see each other they made pennants and flags.’ Gongs, drums, pennants, and flags are the means to unify the men’s ears and eyes. When the men have been unified the courageous will not be able to advance alone, the fearful will not be able to retreat alone. This is the method for employing large numbers. /.../ Thus in night battles make the fires and drums numerous, and in daylight battles make the flags and pennants numerous in order to change the men’s ears and eyes.’

(Written around 520 BC by Sun-tzu, pp. 77-78, translated 1996 by Sawyer, R.D.)

Observe that Sun-tzu advocates a very narrow interface in the army; just four parameters: gongs, drums, pennants, and flags. In order to be able to command an army during combat, the interface must also have been quite shallow (i.e. have a limited set of commands that could be relayed through the interface).

The partitioning of states is the limiting of attribute ranges. This strategy does not exclude attributes from the model, but limits the values they can assume. In practice, this means that several nearby states in the studied system replace one state in the model.

By standardising the interface of the encapsulated system, the system itself may be replaced by a different system as long as it displays the same interface.

By encapsulating behaviour in a system, so that only a narrow interface remains accessible to an outside observer, a Black Box is created. The only way for an outside agent to manipulate the state of the box is through the inbound interface. The behaviour of the box can only be observed through the outbound interface. All the internal mechanisms and algorithms within the box are invisible from the outside.

The Black Box as a model was introduced by Ashby (1956). Its name is derived from the field of electrical engineering, where boxes containing electronic circuitry at the time often were painted black (Geyer, 1995).

Beer (1959) proposes a strategy that he calls “Completion From Without”. A Black Box is inserted into a regulated system. This box contains algorithms of a higher order than those of the system itself, thereby enabling the system to “break out” of logic traps such as Gödel’s Incompleteness Theorem (“What this sentence says cannot be proved.”). The Black Box can be seen as an “escape clause”, or Deus ex machina, which represents adjoining hierarchical levels where function is encapsulated. This is consistent with the statement by Checkland and Scholes on page 53 where they state that it is impossible to explain a system from within.

By organising subsystems hierarchically, interrelations and “parent-child” relationships can be explored. When studying a complex system, a hierarchic model helps choosing abstraction level.

“By a hierarchic system, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem.”

(Simon, 1962, p. 458)

By using hierarchy in bringing order to the world, it becomes possible to isolate elements that are on the same hierarchical level or in the same vertical branch and treat them in the same

---

9 “The Art of War” was written 2500 years ago and is still used in military training around the world. In recent years, is has been used in marketing as well as other areas.
The use of hierarchy is one of the most fundamental principles in scientific modelling.

### 3.3.2 Canonical models

A working group within NEVEM (Netherlands Association for Logistics Management) has elaborated on the hierarchical encapsulation (NEVEM, 1989) and proposes a model of nested “black boxes”, as can be seen in Figure 24 below. The figure is a mix of the two kinds of system description stated by Rosen on page 56 (Rosen, 1972). It is an internal description in that the planning and control processes are visible, and an external in that the levels are nested and only “visible” through their interfaces.

![Figure 24 Processes on various levels (redrawn from NEVEM, 1989).](image)

Casti has a similar model of the relations between input and output, although with a mathematical approach.

![Figure 25 Casti's model of the input/output relation (redrawn from Casti, 1989, p. 109)](image)
Casti explains that the state space, X, must be completely observable as well as completely reachable in order to ensure that a regulator that wants to accomplish its goal must have a canonical model of the system it regulates. The canonical form enables a maximum level of description (completely observable) of the function of the modelled system as well as displaying the highest degree of transparency (completely reachable).

"Assume that we are given any model \( \Sigma = (X, g, h, x_0) \). Then we say that \( \Sigma \) is completely reachable if for any state \( x^* \in X \), there exists an input sequence \( \omega \in \Omega \) and a time \( T \) such that \( x_t = x^* \), i.e., the input \( \omega \) transfers the system state from \( x_0 \) at time \( t = 0 \) to \( x^* \) at time \( T \). We call \( \Sigma \) completely observable if any initial state \( x_0 \) can be uniquely determined from knowledge of the input sequence, together with observation of the system output \( y \), over an interval \( 0 < t < T \). Putting the two concepts together, we call \( \Sigma \) canonical if it is both completely reachable and completely observable."

(Casti, 1989, pp. 113-114)

What Casti explains is that any model is completely reachable if any state within the state space can be obtained through the inbound interface. Likewise, the model is completely observable if its initial state can be derived from the output together with the input, i.e. a Black Box where every initial state can be explained by studying its interfaces.

The model from NEVEM (1989), see Figure 24 on page 61, illustrates how a description depends on the level of abstraction. An observer on level (b) will regard level (a) as the environment and level (c) as a Black Box. According to Casti above, a model is canonical if its state space is fully observable as well as reachable. By necessity, the definition of a canonical model must be recursive. If the state space includes the states of subsystems as well, although they are Black Boxes, they too must be reachable and observable. If a model of a system would be canonical on all levels, it would mean that an observer might choose where to set the level of abstraction, that he has sufficient knowledge of the system to place himself on any level and view the system from there. Mind that this is not reductionism. The levels of abstraction let the observer choose what will be the environment and what will be the Black Box (the encapsulation). The holistic perspective of systems science still applies, since a “truth” gained at one level might not be valid or applicable to the next. Thus, in a canonical model, an observer can move between levels of abstraction at will, and for each level gain as much information about its internal function as possible, as well as study the interfaces of the subsystems.

### 3.3.3 To control a system – Cybernetics

The Regulator is a central object in systems theory. A regulator is capable of manipulating a Black Box towards a goal state, given that it (the regulator) has sufficient variety. The theories on system control are mainly concentrated within the branch of system theory called cybernetics. The word cybernetics comes from the Greek word kybernetes, meaning steersman and is defined by Wiener as “…/ the entire field of control and communication theory, whether in the machine or in the animal /…/” (Wiener, 1961, p. 11). Ashby states further: “What cybernetics offers is the framework on which all individual machines may be ordered, related and understood.” (Ashby, 1956, p. 2). Beer comments cybernetics thus:

---

10 Some explanations to the symbols in this model:
X is the total state space for the model, g represents the inbound interface, h represents the outbound interface, and \( x_0 \) is the initial state of the model. \( x^* \) is any state within the state space and \( \Omega \) is a set of inputs.
“/…/ Wiener must take the final responsibility for the currency of this ugly word {cybernetics}, and also the credit for its aptness.”

(Beer, 1959, p. 30)

There are, according to Ashby, three core parts of cybernetics: Mechanism, Variety and Regulation and Control. Mechanism means the study of machines, real or imaginary, and how they behave. Variety can be used to indicate the complexity of a system, and as Ashby states in his Law of Requisite Variety: “/…/ only variety can destroy variety” (Ashby, 1956, p. 207), meaning that a system with large variety (i.e. a system that can assume a large number of states) only can be controlled if the regulator has equal or larger variety. Regulation and Control encompasses such a notion of an external regulator to “destroy a system’s variety”. There are several models where a regulator of this kind is used, for instance by Beer (1959) and NEVEM (1989).

The Law of Requisite Variety is a law of nature. The variety of a system (the number of states it can assume, i.e. the number of states within its state space) decides how a large the variety of a regulator must be in order to maintain full reachability (as defined in respect of canonical modelling, see Chapter 3.3.2).

3.4 Object-orientation

Object-orientation is a collection of methods originally developed to design highly maintainable and modular computer applications. Booch (1991) – one of the pioneers on the subject – states that object-orientation is based upon seven principles:

“Object-oriented design is built upon a sound engineering foundation, whose elements we collectively call the object model. The object model encompasses the principles of abstraction, encapsulation, modularity, hierarchy, typing, concurrency and persistence. By themselves, none of these principles are new. What is important about the object model is that these elements are brought together in a synergistic way.”

(Booch, 1991 p. 25)

Booch states further (1991 p. 38) that object-orientation, or as he calls it – the object model, consists of four major elements (abstraction, encapsulation, modularity, and hierarchy) and up to three minor elements (typing, concurrency, and persistence). The major elements are required for a model to be object-oriented.

As recognised from Chapter 3.3, the major elements are all parts of the simplification strategies that can be used when modelling complex systems. The minor elements are more related to the computer programming domain. Although none of them are in conflict with the foundations of systems science, the uses of them are most apparent when designing computer software.

Concurrency means that several parts of a model can act independently of one another. In a general system, this is implicitly assumed. In a computer programming, however, concurrency is not always taken for granted, hence the need to explicitly define it (Yourdon, 1994, P. 10).

Persistence means that objects and their states can be stored and retrieved at a later time. In programming, this requires file handling, database structures and so on. Persistence is one of the minor elements and will not be applied to transport systems since the modelling of physical processes naturally implies that objects continue to exist after the study of them is finished.

There are, of course, several definitions of object-orientation, or OO. Some of them are more focused on programming and some on general modelling issues. Rumbaugh et al., for
instance, describe object-orientation as “/…/ a way of thinking abstractly about a problem using real-world concepts /…/” (Rumbaugh et al., 1991, p. 1). They then relate object-orientation to their core subject – programming:

“Superficially the term ‘object-oriented’ means that we organize software as a collection of discrete objects that incorporate both data structure and behaviour. /…/ [characteristics of an object-oriented approach] generally include four aspects: identity, classification, polymorphism, and inheritance.”

(Rumbaugh et al., 1991, p. 1)

Yourdon emphasises the encapsulation into interacting subsystems as instrumental for object-orientation:

“A system built with object-oriented methods is one whose components are encapsulated chunks of data and function, which can inherit attributes and behavior from other such components, and whose components communicate via messages with one another.”

(Yourdon, 1994, p. 2)

There are several important concepts and specific areas within the scope of OO, such as classification, encapsulation, polymorphism, hierarchy etc. The rest of this chapter is devoted to some of these areas within the context of this thesis. In Appendix 2, the basics of OO are described in more general terms.

3.4.1 Classes and objects

Central in object-oriented methods is the use of the two concepts objects and classes.

“An object has state, behavior, and identity; the structure and behavior of similar objects are defined in their common class; the terms instance and object are interchangeable.”

(Booch, 1991 p. 77)

Classes act as templates for objects. The definition of an object above closely resembles that of a Black Box in cybernetics (see for instance Ashby, 1956, pp. 86-93; Beer, 1959, pp. 49-57). Booch argues that a model of a complex system, based on objects that are part of a class hierarchy is canonical.
In the figure above, the classes (C₁ to C₇) are described and related to each other. The system that is described using these classes consists of eight subsystems (D₁ to D₈). Each of these subsystems contains objects based on one or more of the classes. The relations defined in the class diagram are transferred to the objects. The class diagram acts as a “blue-print” repository for the real-world system.

Objects have four defining characteristics (from Rumbaugh et al., 1991): identity, classification, polymorphism, and inheritance.

Identity
Data is divided into discrete, distinguishable entities called objects, for instance a pallet or a trade agreement (objects can be conceptual and non-physical). Each object is unique, even if they share the same attributes. Booch says:

“Identity is that property of an object that distinguishes it from all other objects.”

(Booch, 1991, p. 84)

According to Rumbaugh et al., identity means that “...objects are distinguished by their inherent existence and not by descriptive properties that they have” (Rumbaugh et al., 1991, p. 22). Both these definitions state that the identity of an object is not a property, an attribute, which is part of the state of the object; it is rather so that the object exists and occupies space, which, metaphorically speaking, means that it is a unique entity.

Classification
Objects with the same data structure (attributes) are grouped into a class. Each class describes a possibly infinite set of individual objects (Rumbaugh et al., 1991 p. 2). Each object is a separate instance of the class, for example Pallet A and Pallet B (both are pallets but they are different entities with different identities).
Polymorphism

The same operation may behave differently on different classes. For example, the operation “fly” is different when executed on a kite or an aircraft, even though it means the same thing. Each object has inherent knowledge of how to perform its own operations (Rumbaugh et al., 1991 pp. 2-3).

Inheritance

Attributes and operations are shared between objects based on a hierarchical relationship. Objects can be refined and factored into subclasses where each subclass inherits the attributes of its parent, or superclass (Rumbaugh et al., 1991 p. 3). The boxes on a pallet, for example, inherit the geographical position of the pallet itself.

3.4.2 Attributes and operations

The object as a concept is closely related to the Black Box in cybernetics. As with a Black Box, the object has input (operations) and output (attributes). An object is based on a class, and classes can be seen as blueprints for objects with the same data structure.

![Figure 27 A class is a template for real-world objects. This is the class Pallet that represents all objects that share the same data structure. The topmost cell contains the name, the middle cell contains the attributes, and the bottom cell the operations.](image)

As seen in the figure above, a class consists of a name, a vector of attributes, and a vector of operations. The name is used to identify the class when building and analysing models.

Booch defines the state of an object:

“The state of an object encompasses all of the (usually static) properties of the object plus the current (usually dynamic) values of each of these properties.”

(Booch, 1991 p. 78)

As with a system, the state of an object is the collective value of all its attributes at a given time. Thus, the state can be expressed as a vector, \( \mathbf{S}_t = \left[ a_{1,t}, a_{2,t}, ..., a_{n,t} \right] \) where \( a_{i,t} \) is the value of attribute, \( i \), at the time, \( t \). As has been shown earlier in Chapters 1.2.2 and 3.1, a transport is a transformation between the states \( \mathbf{S}_{t_0} \) and \( \mathbf{S}_t \).

There is only one way to change the state of an object deliberately and that is by executing at least one of the operations that the object gives access to, either by itself or by inheritance from other objects. The effect of executing operations is called object behaviour:

---

11 In this thesis, the "properties" of an object are called "attributes"
“Behavior is how an object acts and reacts, in terms of its state changes and message passing.”

(Booch, 1991, p. 80)

The behaviour has an analogy in systems science: the transformation. A transformation converts input into output, just as an operation on an object. An operation is capable of changing the values of attributes, performing data manipulation and returning results. In the computer-programming world, the execution of an operation takes (processor-) time, in the real world each operation takes time and can consume resources, thereby generating cost. In essence, the operations of an object collectively form a “metabolism”-mechanism of sorts where “energy”-input is transformed into “work”.

An object can have more than one operation, forming a vector of available operations:

\[ O = \{ o_1, o_2, \ldots, o_n \} \]

Examples of how operations can be represented in a model (examples are from the Pallet class in Figure 27):

- LiftUp(Distance) executes an operation that lifts the pallet just enough to clear the ground, thus putting the object into such a state that it can be moved.
- MoveTo(newX,newY,newZ) executes the MoveTo operation with the new position (goal state) as user input. The operation changes the Pos X, Pos Y and Pos Z-attributes and thereby the state.
- PutDown()\(^12\) executes an operation that lowers the pallet to the ground.

Consider once again Figure 29. Since the only way to change states is to apply one or more of the available operations in sequence, the transportation of a pallet in a warehouse to the loading area and the subsequent loading of the pallet onto a truck would at least require the following operations:\(^13\)

- Pallet(i).LiftUp (10 cm)
- Pallet(i).MoveTo(X_{\text{LoadingArea}}, Y_{\text{LoadingArea}}, 0)
- Pallet(i).MoveTo(X_{\text{Truck}}, Y_{\text{Truck}}, Z_{\text{Truck}})

In a complex environment with large variety, the desired end state can often be accomplished in a number of different ways. Each solution is a unique series of operations to execute, thereby generating different trajectories.

---

\(^{12}\) The double parentheses “()” after the operation serve two purposes in OO: 1. they signify that it is an operation and not an attribute, and 2. they may contain arguments that need to be passed to the operation. They have their background in computer programming where parentheses often are used to contain function arguments.

\(^{13}\) To signify that it is an object, the notation Pallet(i).Operation is used, where \(i\) represents the identity of an individual pallet. The dot “.” between Pallet(i) and Operation is a common method in programming to address a member of a collection. In this case it is a member of the vector of operations on Pallet(i).
Let us add another class to the model:

<table>
<thead>
<tr>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>Height</td>
</tr>
<tr>
<td>Depth</td>
</tr>
<tr>
<td>X-pos</td>
</tr>
<tr>
<td>Y-pos</td>
</tr>
<tr>
<td>Z-pos</td>
</tr>
<tr>
<td>Consignor</td>
</tr>
<tr>
<td>Consignee</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Open</td>
</tr>
<tr>
<td>Close</td>
</tr>
<tr>
<td>MoveTo(X,Y,Z)</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

**Figure 28** The class Box.

When boxes are put on the pallet, a number of their attributes become locked (the X, Y and Z-positions are inherited from the pallet). The consignor is probably the same for all boxes as well (consignor can of course also be expressed in coordinates).

In the model, the case of boxes on a pallet would look like this:

<table>
<thead>
<tr>
<th>Pallet</th>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>Weight</td>
</tr>
<tr>
<td>Height</td>
<td>Height</td>
</tr>
<tr>
<td>Depth</td>
<td>Depth</td>
</tr>
<tr>
<td>X-pos</td>
<td>X-pos</td>
</tr>
<tr>
<td>Y-pos</td>
<td>Y-pos</td>
</tr>
<tr>
<td>Z-pos</td>
<td>Z-pos</td>
</tr>
<tr>
<td>LiftUp(Height)</td>
<td>MoveTo(X,Y,Z)</td>
</tr>
<tr>
<td>PutDown</td>
<td></td>
</tr>
<tr>
<td>MoveTo(X,Y,Z)</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 29** An object of the class Pallet can contain several objects of the class Box. The denotations 1 and * means that one pallet (1) may contain many (*) boxes.

In order to move one of the boxes (Box “j”), the operation may look like this: Pallet(i).Box(j).MoveTo(X,Y,Z). By invoking the hierarchic relationship between Pallet(i) and Box(j), the regulator does not have to keep track of the individual attributes for Box(j). If it already knows the spatial coordinates for Pallet(i), the coordinates for Box(j) are inherited and do not have to be stored, thereby reducing variety.

In exactly the same way, an object from the class TerminalFacility can contain a multitude of individual Pallet objects etc. In the terminal facility, the contained objects all display different states, but some of the attributes are locked to the parent object (X-Pos, Y-Pos, Z-Pos etc.), so even if all the attributes have different values, they are all within a specified range defined by the terminal’s width, depth and height. This means that a parent object can enforce boundary constraints upon its children, making them obey certain fundamental system rules (same as with the pallet and its boxes above).
3.4.3 Object-orientation and systems science

As has been explained earlier, a reductionist approach cannot explain a system adequately, since the emergent properties are observable when, and only when, the various parts are combined and allowed to interact, thus forming a system. This leads to an important question concerning the relation between object-orientation and systems science:

Are systems, modelled using object-orientation, really systems according to the definitions stipulating emergent properties as a prerequisite for a system? Are they not simply a collection of algorithms and rules? Do the parts truly interact?

If one should ask a computer programmer, the answer would in most cases be "no" (he means that object-oriented models are exactly the sum of the individual parts - no more, no less). This most probably is a direct result of a predominant reductionist view in that the function of the entire universe can be broken down to elementary physics (and thereby predicted) if only the model is detailed enough. However, as was stated earlier, the reductionist view depends on one important assumption: there can be no interaction between the reduced model and its environment. It is the view of this thesis that interaction between objects is not only allowed in OO, it is counted upon. Object-oriented models can be considered system models and they display emergent properties that are not present when studying the objects individually.

Caution is needed, though. An object-oriented model requires enumeration of attributes and operations. If not treated properly, this can lead to a mechanistic view. One important distinction between an object model, designed to produce software, and a model made of a real-world system is that the software model is finite in variable range and that the real-world model may contain open-ended variables. Just because a modeller is required to name attributes and operations does not mean that they are finite or discrete, neither in width, nor in depth. The principle of abstraction allows for delimitations and idealisations. The mechanistic view is related to the mathematical side of cybernetics as described by, for instance, Ashby (1956) and Wiener (1961). An object-oriented model of a system can be described by using equations as in cybernetics, it can be described graphically as in for instance UML, or it can be described in algorithms (computer code). There is no real difference between the description languages per se, although the graphical language of object-oriented methods is an efficient abstraction in itself.

Systems theory in general, and especially cybernetics, shares several concepts and terms with object-orientation. The theoretical field of object-orientation is not well-known for its contribution to the philosophy of science. Even though object-orientation is a frequent subject in computer science journals, it is often discussed in the perspective of the programmer. Object-orientation is more than a family of methods, however. The similarities to the area of systems science are numerous. Some of the most prominent analogies are shown in Table 9 below.
It is evident in the literature that system theory has influenced the ideas behind object-orientation. The development has been focused on programming languages and database designs, and methodological reflections are not common. However, Booch, as one of the most noteworthy “OO-writers”, says:

“Actually, the object-model has been influenced by a number of factors, not just object-oriented programming. /…/ the object model has proven to be a unifying concept to computer science, applicable not only to programming languages, but to the design of user interfaces, databases, knowledge bases, and even computer architectures. The reason for this widespread appeal is simply that an object orientation helps us to cope with the complexity inherent in many different kinds of systems.”

(Booch, 1991, p. 33)

According to Booch, object-orientation is founded partly on an idea of a generality in modelling. He also explicitly states that systems’ complexity can be handled with the help of OO. As shown earlier (in Chapter 3.3.1 on page 59), systems’ complexity can be reduced by one or more simplification strategies. Four of the most important are:

- Exclude variables
- Partition states
- Break down into subsystems
- Organise subsystems hierarchically

Each of these strategies can be addressed by using OO-based modelling. When constructing an OO-based model, the abstraction level is adjusted according to the model-builder:

<table>
<thead>
<tr>
<th>Systems theory</th>
<th>Object-orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>System model ↔ Object model</td>
<td></td>
</tr>
<tr>
<td>State ↔ State</td>
<td></td>
</tr>
<tr>
<td>Outbound (display) interface ↔ Attributes</td>
<td></td>
</tr>
<tr>
<td>Inbound (control) interface ↔ Operations</td>
<td></td>
</tr>
<tr>
<td>Black Box ↔ Object</td>
<td></td>
</tr>
<tr>
<td>Trajectory ↔ Path in Statechart</td>
<td></td>
</tr>
<tr>
<td>Encapsulation ↔ Encapsulation</td>
<td></td>
</tr>
<tr>
<td>Abstraction ↔ Abstraction</td>
<td></td>
</tr>
<tr>
<td>Hierarchic architecture ↔ Hierarchic architecture</td>
<td></td>
</tr>
<tr>
<td>Transformation ↔ Object behaviour</td>
<td></td>
</tr>
</tbody>
</table>
“An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus crisply defined conceptual boundaries, relative to the perspective of the viewer.”

(Booch, 1991, p. 39)

Therefore, the exclusion of variables is performed embedded within the model-building process itself. The process of classification is exactly the same thing as the partitioning of states, where system states that resemble each other are grouped into a separate partition. The use of subsystems and hierarchic structure is also inherent in OO-models.

Based on this reasoning, it seems likely that OO can be used to build “good” models of complex systems according to the criteria stated by Casti on page 58.

3.4.4 Types of object models

As was described in Chapter 3.1.2, models of transportation and logistics systems can be of one of the three types relational, chronological, or snapshot models. As seen in Table 10 below, the OO-diagrams can also be classified into these types:

*Table 10 The various OO-diagrams and their corresponding model type*

<table>
<thead>
<tr>
<th>Model type</th>
<th>OO-diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational</td>
<td>Class diagram</td>
</tr>
<tr>
<td></td>
<td>Use case diagram</td>
</tr>
<tr>
<td>Chronological</td>
<td>Sequence diagram</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Statechart</td>
</tr>
</tbody>
</table>

The class-, and use case diagrams are relational models that describe structure and relationships between entities. The sequence diagram describes events in time, and the statechart describes snapshots of the system at various points in time.

The object-oriented framework is in itself a relational model where diagrams describing various views of the same system are related to each other through a common set of abstract entities – objects.
3.5 Object-orientation applied to transportation and logistics

Since the emergence of the object-oriented programming languages, several additional application areas have been explored. The application on the domains of transportation and logistics is not new. What is regarded as the first object-oriented programming language, SIMULA, was developed specifically for writing discrete-event simulation models (Banks et al., 1996, p. 133).

A total of 37 (mostly journal) papers were found using object-orientation in connection with logistics or transportation. A full list of the papers together with their classification into application areas and modelling approaches can be found in Appendix 4. All of the papers have key words or titles containing “object-orientation” or similarly combined with terms related to the application areas above. The literature review is summarised in Table 11, below. Of these papers, 107 different combinations of application areas and modelling approaches were observed.

Table 11 A summary of the application areas and modelling approaches of object-orientation on transportation and logistics found in the literature. Each cell corresponds to a specific combination of application area and modelling approach, and shows the number of occurrences of this combination in the found literature. One source may contain several areas and approaches and may therefore contribute to several cells in this table.

<table>
<thead>
<tr>
<th>Application areas</th>
<th>Modelling approaches in the studied papers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Design</td>
</tr>
<tr>
<td>A Transportation</td>
<td>16</td>
</tr>
<tr>
<td>B Manufacturing</td>
<td>16</td>
</tr>
<tr>
<td>C Automation</td>
<td>6</td>
</tr>
<tr>
<td>D Public transport</td>
<td>2</td>
</tr>
<tr>
<td>E Warehousing</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>45</td>
</tr>
</tbody>
</table>
3.5.1 Application areas and modelling approaches identified

The following areas where object-orientation is used within the transportation and logistics domain have been identified in the papers:

Table 12 Identified application areas of object-orientation within the domain of transportation and logistics

<table>
<thead>
<tr>
<th>Area</th>
<th>Uses of object-orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transportation</td>
<td>Object-orientation is used in modelling arbitrary transportation systems</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Object-orientation is used to study or design manufacturing systems</td>
</tr>
<tr>
<td>Automation</td>
<td>Automated processes of any kind, i.e. AGV transports or automated manufacturing are studied and controlled using object-oriented foundations.</td>
</tr>
<tr>
<td>Public transport</td>
<td>Object-orientation used mainly when designing, controlling, or analysing a public transport system.</td>
</tr>
<tr>
<td>Warehousing</td>
<td>Object-orientation is used to analyse, control, or design warehousing operations.</td>
</tr>
</tbody>
</table>

Also, in the studied papers, the motives for the OO-based modelling are different. Four main modelling approaches have been identified:

- Modelling approach 1: The use of OO-based modelling to design new systems
- Modelling approach 2: The use of OO-based modelling to design control mechanisms for an existing system
- Modelling approach 3: The use of OO-based modelling to design simulation models of an existing system
- Modelling approach 4: The use of OO-based modelling to analyse an existing system

Modelling approach 1 differs from the other three. With this approach, a real-world system is designed using object-orientation (see Figure 30 below).

Figure 30 Modelling approach 1 – design of real-world system using object-orientation.

Examples of this approach can be found in Kovács et al., where manufacturing systems are designed using object-oriented technology, or in Yun and Choi where a simple object model is used in the design of a container terminal (Kovács et al., 1999; Yun and Choi, 1999).

In Figure 31 below, the other three approaches are illustrated. They all originate in a real-world system, which is modelled using object-orientation. This model system can be used for
three different purposes. In modelling approach 2, the system model is used to design a control system. This control system is used to regulate the real-world system. Examples of approach 2 can be found in Betlem and van Aggele, where a production system is controlled by a set of rules designed using an object model of the production system (Betlem and van Aggele, 1994).

![Diagram of modelling approaches 2, 3, and 4. An object model of a real-world system used for three purposes: control, simulation, or analysis.](image)

The third approach, to use an object model in simulation, is close to a standard in discrete-event simulation. Paolucci and Pesenti, for instance, present an object-oriented simulation of an underground railway system, where the simulation model produces results that are used to create new time-tables (Paolucci and Pesenti, 1999). The use of OO-based simulation models is widespread and lies outside the scope of this thesis. Therefore, the number of papers, using only this approach, is greater than presented here. The ones in the review were selected because of their use of at least one of the other approaches, i.e. they used the OO-based model for another purpose than just creating a simulation model.

Modelling approach 4 is adopted by Wache when using Object-Oriented Analysis to gain understanding of a flow of materials (Wache, 1998).

### 3.5.2 Conclusions of review

Modelling approaches 1 and 2 are in majority. Approach 1, object-orientation used to design systems, is mainly used in transportation and manufacturing. The transportation area is clearly
focused on the design approach, whereas the manufacturing area is equally divided between the design of systems and the construction of control systems. Many of the studied papers use more than one approach, such as 1 and 2 (19 papers, some of them including additional approaches). 24 of the papers focused on one single application area, 12 papers had 2 areas and one paper had 3. There is a clear gap in approach 4, the analysis of systems using object-orientation. Even though the area of transportation constitutes more than half of the papers using approach 4, this only amounts to 9 papers, ranging from 1995 to 2004.

According to the summary presented in this section, there are several works that use object-orientation to either design or to control existing systems in the transportation domain. This thesis uses object-orientation to:

1. Analyse transportation systems (approach A-4).
2. Propose a framework for the design of transportation systems (approach A-1).
3. Make use of the principles behind object-orientation when controlling systems (approach A-2).

The last approach (A-2) is only used in a conceptual way, and no actual control systems are designed, although principles for them are presented based on the case studies.
4. BUILDING A FRAMEWORK

This chapter contains deeper discussions on the theories from the frame of reference and how they can be extended to encompass the problem domain of this thesis.

Although object-orientation can be considered “applied systems science”, some aspects of it may, however, still be considered as outside the scope of systems science, such as the activity of programming. The notion of transportation as a systems science is also subject of discussion. In this thesis, a systems perspective is adopted, although a transportation chain may be regarded with for instance a mathematical and quantitative approach, e.g. the Travelling Salesman problem. These mathematical methods, often used by operations researchers, serve to solve a different set of problems than those associated with the systems perspective. Also, object-orientation is often used when applying quantitative methods to transportation systems. Object-oriented modelling is an excellent tool to use when designing large optimisation problems, since the construction of the algorithms closely resembles programming. The triple overlap between the areas of systems science, transportation, and object-orientation is the theoretical focus in this thesis. This chapter will therefore discuss the various aspects of object-oriented systems models of transportation chains.

4.1 A regulator model of a Transport Control System

This section is dedicated to the development of a conceptual model of a regulator capable of altering state for goods objects. A brief description of the model is followed by more detailed account of its different parts.

4.1.1 The model

The conceptual model of the transportation system described on page 4 consists of the following elements:

- A transportation system containing a state space for goods objects
- A transformation process resulting in a trajectory
- Goods in two states, before and after trajectory
- Resources used in the transportation system to complete the trajectory
The model is shown in Figure 32 below:

![Diagram of the conceptual model of a transportation system]

**Figure 32 The conceptual model of a transportation system.**

The theories of cybernetics and object-orientation are applied to the conceptual model to form an extended model. The extended model consists of a regulator and goods objects (before and after regulation). The regulator is synonymous with the transportation system in the conceptual model. The regulator has a number of resources at its disposal. The resources can be used when the TCS interacts with the goods objects. The goods objects display two interfaces, attributes and operations (outbound and inbound) as well as a goal state (desired attribute vector). The regulator has the ability to alter state for objects of the Goods class. The regulator is also able to interpret the state of objects of the Goods class and execute appropriate operations.

**Attributes**

\[ \overrightarrow{A} = [A_1, A_2, \ldots, A_k] \]

**Operations**

\[ \overrightarrow{O} = [O_1, O_2, \ldots, O_n] \]

**TCS**

A regulator capable of achieving state transitions for objects of the goods class

**Resources**

\[ R_1, R_2, R_3, \ldots, R_m \]

**Goal:**

\[ \overrightarrow{A'} = [A'_1, A'_2, \ldots, A'_k] \]

**Figure 33 Conceptual model of a regulator capable of changing state for goods objects.**

This model has explicit relations to previous work, as well as general affinity with the areas of object-orientation, transportation and cybernetics. The following sections will briefly account for the explicit influences.

The model is derived from the concepts of object-orientation. Several items in the model can be related to object-oriented concepts, such as

- The Goods object
- Attributes
- Operations
- The regulator. An object of a higher order.
The object-oriented methods are described in more detail in Chapters 2.2.2, 3.4, and Appendix 2.

4.1.2 The modelling of trajectories

A trajectory in the real world is multi-dimensional. A model of a transformation process can be seen as a projection of the “real” trajectory. This projection is an abstraction where only a few attributes are modelled, for instance geographical coordinates. The rest of the attribute vector is disregarded. For a transport, the trajectory contains states where geographical positions (and probably several other attributes as well, even if they are not modelled) change.

This traditional way of modelling a transportation chain is well-founded and understandable. Arrows describe the sequence of nodes and links. It is easy to follow the different states the goods must pass, i.e. its trajectory, in order to reach its goal state. The regulator model takes a new perspective on this. Focus is on the transitions between states, rather than the states themselves. These transitions are always the result of manipulation of the goods objects through their inbound interfaces. This manipulation is performed by a TCS (there can be different TCSs for each transition step in the trajectory). Compare the hierarchical composition of regulated state transitions in Figure 34 below with the hierarchical structure of a network (presented in Figure 15 on page 42).

![Hierarchical TCS-structure](image)

*Figure 34* The trajectory in the network model above is represented by the different attribute vectors (states). Each interface node/link is represented by a state. In the TCS model, the focus is on the transformations between states, not on the states themselves.

Paramount in the model is the object of the Goods class. The Goods superclass has a few specifics:

- It always contains attributes for geographical location (X, Y, Z). Since an object is a physical entity, it has physical coordinates.
- There are always one or more goal states for each object, in some way related to geographical position. In order to be labelled “Goods” there must be a desired set of attribute values including geographical position.
4.1.3 Demands on a TCS

In order for a TCS to be successful it has to, according to the Law of Requisite Variety, possess equal or greater variety than the system it regulates. Consider an object of the Goods class. The object has a number of attributes, and each attribute may adopt several different values. The variety for the object equals the total number of different states (attribute-value combinations) that can be assumed. Being a goods object, it also has one or more goal states. To achieve one of these states, values for a number of the attributes must change. To facilitate these changes, the object presents an inbound interface consisting of operations. A successful TCS must be able to:

1. Interpret the object’s goal state
2. Understand the inbound interface to such extent that the goal state, theoretically at least, can be achieved
3. Perform the right operations in the right order and in accordance with constraints to achieve the goal state

Next, consider the addition of another goods object of the same class (displaying the same inbound as well as outbound interface). The only thing that makes the two objects different is their initial states and their goal states. The TCS needs a certain capacity to handle several objects at once, but apart from that, no other demands are given the regulator. The next object to be regulated is also a goods object, but of a different class. The structures of the inbound and outbound interfaces differ from the previous objects. The TCS needs to understand these new interfaces as well, in other words it needs new knowledge. Variety as such does not provide any information about how a state change can be achieved. As discussed in section 1.2.7, Heylighen argues for a Law of Requisite Knowledge that applies to a control system. This adds a planning element to the TCS that is not addressed by the Law of Requisite Variety.

"In order to adequately compensate perturbations, a control system must "know" which action to select from the variety of available actions."

(The Law of Requisite Knowledge according to Heylighen, 1996)

Variety is an aggregated concept and is (for transportation systems) at least four-dimensional as a function of object attributes, object operations, TCS knowledge and TCS capacity. For each state change, the TCS chooses from several alternative operations to reach a certain state. This choice is not random but depends on the TCS’s knowledge of the regulated object’s state space as well as knowledge of its own capacity needs in relation to other objects that are regulated simultaneously.

\[ \text{TCS variety} = f (\text{Object attributes, Object operations, TCS knowledge, TCS capacity}) \]

4.1.4 Capacity of the TCS

Capacity is a well known term in transportation. It depicts the extent to which a resource can be used.\(^{14}\) Capacity is measured in units that are inherited from the resource it encompasses. A 40 ft container has a capacity of approximately 30 metric tonnes; a normal car has a capacity of 5 persons, and so on. A regulator in a transportation system, for instance a TCS operating a

\(^{14}\) Capacity: “the potential or suitability for holding, storing, or accommodating” or “the maximum amount or number that can be contained or accommodated”

(Merriam-Webster, 2006)
terminal facility, also has a capacity. How this capacity is measured varies from case to case. In some cases, the capacity can be transformed into time, i.e. the time needed/available to regulate all the objects currently under the control of the regulator. In other cases, capacity is measured in physical terms, as in the number of unoccupied slots in a warehouse.

These two types of measurements of capacity can be called consolidation and regulation capacity. Consolidation capacity gives the number of objects that can be encapsulated within the system (e.g. the number of pallet slots in a container). Regulation capacity is the number of objects that a TCS can regulate at the same time (e.g. the number of pallets in a terminal).

4.1.5 Knowledge of the TCS

The TCS must possess, besides capacity, the ability to regulate the objects under its control. This ability, here named knowledge, is a measurement of how compatible the TCS and the regulated object are. According to Ashby and Conant, the best regulator is the one that is a model of the system it regulates, i.e. when the TCS and the system are homomorphic.15

In a transportation system, the lack of requisite knowledge may for instance manifest itself when a container arrives at a post office terminal. The terminal facility is quite capable of handling the interfaces that are common to the system, such as letters and parcels. The facility does not have any handling equipment capable of moving a container weighing 30 tonnes or more. So, even if there is room for the container (i.e. requisite capacity), there is not any inherent knowledge in the system on how to move containers.

4.2 The object as an information processor

When using object-orientation as a description language, it becomes evident that in a transportation system, a majority of the operations that are performed consist purely of the processing of information. In fact, it can be argued that any operation in an object model is a pure information processing operation. A physical transition of an object (for instance a container) is merely a transformation of coordinates to every outside observer that is not actively involved in the transformation process. A Black Box is a processor of input, which yields output. The same can be said of an object, where input in the form of operations yields output in the form of attributes. Dechert supports this claim when he states that a system can be defined by its interfaces:

“The area of contact between one system and another is termed an ‘interface.’ Operationally systems, and subsystems within systems, may be identified by the transactional processes that occur across their boundaries.”

(Dechert, 1965)

By adopting a view of an object as an information processor, the efficiency of the object can be measured using the Partition Law of Information Rates, PLIR (Conant and Ashby, 1970).

15 “Two homomorphic systems have the same basic structure, and, while their elements and operations may appear entirely different, results on one system often apply as well to the other system.”

(Merriam-Webster)
PLIR divides the information processing rates of a system into five parts (from Conant and Ashby, 1970):

Table 13 The five information processing rates of PLIR (Conant and Ashby, 1970)

<table>
<thead>
<tr>
<th>Rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The total rate, F</td>
<td>Defined loosely as “the total amount of computing” within in the system</td>
</tr>
<tr>
<td>The throughput rate, F_t</td>
<td>Measures the difference between input and output</td>
</tr>
<tr>
<td>The blockage rate, F_b</td>
<td>The rate of information that is blocked and not allowed to interfere with the output</td>
</tr>
<tr>
<td>The coordination rate, F_c</td>
<td>The rate of internal communication within the system that allows the system to act as a whole rather than as the sum of its parts</td>
</tr>
<tr>
<td>The noise rate, F_n</td>
<td>The amount of uncertainty per step about the system, given complete knowledge of its input</td>
</tr>
</tbody>
</table>

The total rate F is the sum of the others:

\[ F = F_t + F_b + F_c + F_n \]

According to this model, the processing power that is used for the state transition of an object is F, consisting of throughput, blockage, noise and coordination.

Therefore, if the TCS possesses a canonical model of the object, the throughput rate will be as high as the object itself can allow for. A canonical model means that the TCS knows exactly how to achieve a given state through executing operations. Any losses will be generated within the object and cannot be affected from the outside.

The less compatible a TCS is with the interface of the object, the higher the noise or blockage rates are likely to be. If the TCS has insufficient “knowledge” of the interfaces, i.e. it possesses a non-canonical model of the object, irrelevant data may be passed through the interface. The strategy that a TCS with insufficient knowledge must take is trial and error. Conant also discusses demands on systems design and presents the following guidelines (from Conant and Ashby, 1970):

- produce the minimum allowable output
- perform as little blockage as possible
- reduce internal coordination to the minimum consistent with other requirements
- as far as possible, match components to tasks so that each component is operated at capacity

By applying PLIR, it becomes clear that the state transition of an object can be assessed using the five information rates. In theory, the throughput rate can be measured by an outside
observer, whereas the other four require access to the inner workings of the studied object. However, the throughput rate constitutes the difference between input and output and can therefore be seen as a measurement of the information processing efficiency of the object.

4.3 Encapsulation, constraints, and interfaces

The design of a system is in many cases the design of its interfaces. Since any system can be decomposed into hierarchically nested Black Boxes, a system’s designer is faced with the task of configuring and interpreting interfaces of these boxes.

There are two kinds of interfaces: inbound and outbound. An interface (i.e. a collection of operations and attributes) also displays a depth and a width. The width of an interface represents the number of separate parts it contains, i.e. the number of attributes and operations. The depth represents the variety of each interface part (how many values it can assume). The depth can range from binary (on/off) to continuous. There are always some kinds of constraints that limit the depth to a manageable range of values. Also, the use of encapsulation regulates the width of the interface. It is possible to interpret the depth and width of interfaces graphically (see Figure 35 below).

![Figure 35 Relations between depth and width of interfaces.](image)

When designing a system, there is a trade-off between shallow/narrow and deep/broad interfaces. The further to the upper left in the figure above, the less complicated the interface is, and the less variety is required of a regulator. However, a system like this is not very flexible. The other extreme is a system where nothing is standardised and where all dimensions are continuous. A system like this cannot be easily regulated, since a regulator that can handle variety as large as the one displayed is bound to be more complex than the system itself (the Law of Requisite Variety).

4.3.1 Encapsulation

Interface width depends on the encapsulation. Beer writes:

"Often one hears the optimistic demand "give me a simple control system; one that cannot go wrong". The trouble with such "simple" controls is that they have insufficient variety to cope with variety in the environment. Thus, so far from not going wrong, they cannot go right. Only variety in the control system can deal successfully with variety in the system controlled."

(Beer, 1959, p. 50)
The controllability of a system depends on the interface, i.e. in the variety the interface can transmit. Encapsulation limits this “variety bandwidth”. There are two separate kinds of encapsulation:

- Encapsulation of content. Defines outbound interface.
- Encapsulation of function. Defines inbound interface.

By encapsulating content, a system hides attributes from an observer. Compare this to the simplification strategy of excluding variables discussed in Chapter 3.3.1. This is necessary for any type of interaction with the system to occur, since the effects of outside influence must be observable (see Casti’s definition of complete observability on page 62). The encapsulation of function limits the inbound interface, thereby diminishing the reachability of the system (reachability also defined by Casti on page 62). For regulation of the system to succeed, the interfaces must represent a canonical model of the system with complete reachability and observability for the chosen level of abstraction. A TCS can only achieve states that are observable and may only perform operations that are reachable.

### 4.3.2 Constraints

The reducing of interface depth is accomplished by the application of constraints. The depth is decreased by eliminating possible parameter values for an interface. The variety for a system decreases when constraints are applied. By applying constraints, the degrees of freedom can be kept down and the interface depth be kept shallow.

There are two disjunctive dimensions of a system: static and dynamic. The first encompasses structure, the second, movement. Constraints for a system can encompass both of these dimensions. Both the inbound and outbound interface can be inhibited:

1. Transitory constraints.
   - Limiting operations, i.e. movements, or the transformations. These constraints do not affect the possible states of an object, but rather the operations that are available.
   - Limits the inbound interface

2. Stationary constraints.
   - Limiting states. Constrains the number of possible states for an object. Independent of the transitory constraints.
   - Limits the outbound interface

The transitory constraint limits the dynamics of the system and the stationary limits the static structure.

Constraints can, when adopting a systems perspective, be either hard or soft. Hard constraints are absolute and prohibit any state or operation that may break them. Physical parameters are often hard constraints, such as size of load units etc. A pallet can exist inside a container, but a container can never exist within the geographical limits of a pallet. This is an example of a hard constraint.
The soft constraints are not absolute. They merely inhibit certain states and operations in such ways that they prevent direct regulation. If a warehouse is 100% full, there will be difficulties in adding another item. It is possible, if some of the other items are moved or if a temporary storage is arranged, but the amount of effort is higher than it would have been if the warehouse was not full. These types of constraints are soft.

![Figure 36 Hard and soft constraints and their effects on a TCS.](image)

In the figure above, examples of hard and soft constraints are presented as they can apply to inbound and outbound interfaces respectively.

### 4.3.3 Properties of interfaces

As presented above, interfaces have several properties and can be viewed in several different fashions. In Figure 37 below, interfaces are presented as three-dimensional entities.

![Figure 37 The three dimensions of interfaces: Depth, Width and Direction.](image)

The model above shows how interfaces are treated within the scope of this thesis. Each interface has a direction, i.e. it can be inbound (operations) or outbound (attributes). Each attribute or operation has a depth, i.e. an acceptable value range. For attributes, the depth is the number of values the attribute may assume. For an operation it is the number of states it
may produce. The third dimension is the *width*. Each interface direction has its width, i.e. the number of attributes or operations that are exposed through the interface.

Even though the direction only can assume two values, it is treated as a separate dimension. The reason for this is that the properties of the width and depth dimensions are significantly different with regard to the direction. There is of course the possibility of limiting the direction as well as the width/depth. By not allowing input or output, the system degenerates into a non-interactive entity such as a signpost (no input) or a completely black box, where all the input is absorbed and no output is visible. In Figure 38 below, the different strategies for limiting the interface width/depth is depicted.

![Figure 38 How to enforce limits on an interface.](image)

The interface depth can be reduced by applying constraints. Constraints enforced on the inbound interface are called *transitory*. They limit the number of trajectories in the system state space. Constraints on the outbound interface are called *stationary* and limit the number of states that can be assumed by the system.

The interface width is reduced through encapsulation. Encapsulation on the inbound interface limits the number of exposed operations, i.e. encapsulation of *function*. Encapsulation on the outbound interface limits the number of visible attributes, i.e. encapsulation of *content*.

### 4.4 Controlling the transportation process

When controlling a transportation process, a TCS needs to manipulate the inbound interface of the consignment in such a way that the trajectory ends with the desired goal state. The three complexity types that have been identified (descriptive, computational, and uncertainty-based) represent obstacles that the TCS need to eliminate in order for the consignment to reach the goal state. The control of the transportation process is thus a battle against complexity, waged on three fronts. The control can be divided into three temporal scopes: long-term, medium-term, and short-term.

The *long-term control* mainly consists of designing the TCS and the regulated system in such a way that short- and medium-term control becomes less difficult. The time span when exercising long-term control is measured in years.

The medium-term control is more fine-grained than the long-term control and has a shorter time span (months to weeks). While the outer boundaries of the system are defined through the long-term control, several design aspects can be left to medium-term decisions. This is where most of the planning takes place.
Short-term control focuses on the actual transportation process itself, and what operations to perform. The time span ranges from weeks down to minutes and seconds.

Table 14 The control scopes, their tasks, time spans, and how they affect complexity

<table>
<thead>
<tr>
<th>Control scope</th>
<th>Long-term</th>
<th>Medium-term</th>
<th>Short-term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time span</td>
<td>Years to months</td>
<td>Months to weeks</td>
<td>Weeks to minutes</td>
</tr>
<tr>
<td>Driving questions</td>
<td>What states should the</td>
<td>What components are</td>
<td>What state changes</td>
</tr>
<tr>
<td></td>
<td>system be able to</td>
<td>needed in the system?</td>
<td>should be performed</td>
</tr>
<tr>
<td></td>
<td>assume?</td>
<td>How are the various</td>
<td>and how?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>interfaces designed?</td>
<td></td>
</tr>
<tr>
<td>Important tasks</td>
<td>Design:</td>
<td>Design/operate:</td>
<td>Operate/monitor:</td>
</tr>
<tr>
<td></td>
<td>Define the data-structure</td>
<td>Define actual use cases.</td>
<td>Control the actual</td>
</tr>
<tr>
<td></td>
<td>of the top-level classes</td>
<td>Define interface width</td>
<td>trajectory as it</td>
</tr>
<tr>
<td></td>
<td>of the system, i.e. the</td>
<td>of all classes.</td>
<td>progresses through the</td>
</tr>
<tr>
<td></td>
<td>interface widths.</td>
<td>Apply constraints to</td>
<td>state space.</td>
</tr>
<tr>
<td></td>
<td>Define acceptable data</td>
<td>reduce interface depth.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ranges for these classes,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>i.e. the interface depths.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Define acceptable use</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cases.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity reduction</td>
<td>Descriptive complexity</td>
<td>Computational complexity</td>
<td>Uncertainty-based complexity</td>
</tr>
<tr>
<td></td>
<td>is reduced by robust</td>
<td>is reduced by good planning</td>
<td>is reduced by creating</td>
</tr>
<tr>
<td></td>
<td>design</td>
<td></td>
<td>order</td>
</tr>
<tr>
<td>Examples</td>
<td>Designing terminal</td>
<td>Reserve space on</td>
<td>Loading and unloading</td>
</tr>
<tr>
<td>from the</td>
<td>structure.</td>
<td>vehicles/vessels in</td>
<td>goods.</td>
</tr>
<tr>
<td>transportation</td>
<td>Designing a new route</td>
<td>advance without having</td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>or corridor.</td>
<td>exact knowledge of the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>consignment.</td>
<td></td>
</tr>
</tbody>
</table>

Control by design means that the system that will traverse the desired trajectory is designed in advance so that there will be few alternative trajectories during the actual transportation process. The descriptive complexity can be reduced during the design phase.

In the medium-term control, more detailed design is often present, as well as some direct operations. These operations are not yet associated to any single consignment. They are often concerned with allocation of resources or positioning of unit loads in anticipation of a future transport. In this scope, the computational complexity becomes an issue.

The short-term control can be of two modes; one active – operation, and one passive – monitoring. In the operating mode, the interface of the consignment is manipulated in order to achieve the goal state. Decisions have to be made and more than one alternative trajectory
exists. In the monitoring mode, the consignment is observed as it traverses an already predetermined trajectory. If the consignment deviates from the planned trajectory, the control mode changes to operating. Due to the inherent uncertainty-based complexity that resides in transportation systems, decisions need to be made during the transportation process that would not have been possible to make during the medium- or long-term control scopes.
5. CASE STUDIES

In this chapter, the cases are presented. Each case is described in a separate section.

Each of the separate cases has been chosen because they represent different parts of the same hypothetical trajectory. This trajectory begins with a production process of some kind, yielding an output. The output is subjugated to a warehousing operation; it is ordered and shipped from the warehouse. The transport company has a minimum of two terminals involved, the collecting terminal (A) and the dispersing terminal (B). The consignee receives the goods and in turn delivers them to a warehouse. In Figure 39 below, the model trajectory is shown.

![Figure 39 Model trajectory](image)

Figure 40 below shows how the eight cases are positioned in the model trajectory. The cases have been selected individually so that coverage of all the parts is obtained. Focus lies primarily on shipping (preparing for transportation) and terminal handling in the transportation system.
Figure 40 Referring to the model trajectory, the grey parts are covered in each case. The overall focus is on warehousing, shipping, and terminal operations.

5.1 Case 1 – Transport between two road network terminals

Case 1 is a normal road transport between two domestic terminals. The goods are packaged, ranging from 50 to 1000 kg, and put on pallets for easy handling. A waybill (paper) accompanies each consignment. The goods arrive at the terminal during the day when the local collection lines are operated. In the terminal, the goods are sorted and placed according to zip code at specified loading gates. When a semi-trailer arrives at a specific gate, the
terminal operators load the goods, give the waybills to the driver and seal the semi-trailer. The seal is later broken when the semi-trailer arrives at its destination terminal or when the vehicle is inspected by authorities, who then have to re-seal it. The semi-trailer is driven to its destination terminal, from which the consignments are distributed.

5.1.1 Case objective

This case study – the first one in the project – was conducted to establish a reference base of how a typical road transport system is constituted as well as to establish the data collection method. The case especially studied differences in handling between dangerous and non-dangerous goods. The study encompasses the simplest of systems where the degree of standardisation is high and where goods types are relatively uniform.

5.1.2 Case methodology

The data in this case was mostly collected on a single day and relates to two consignments that were followed from the incoming gates in terminal A to the outgoing gates of terminal B, with the exception of the semi-trailer transport between the terminals. However, the sealing of the semi-trailer at terminal A was observed as well as the breaking of the seal at terminal B.

5.1.3 Actors and roles

This study encompasses two terminals in a line-based distribution network. See Figure 41 below.

![Figure 41](image)

*Figure 41 This case studies a transport between two terminals, A and B.*

This case encompasses two terminals, the sending (collecting) terminal and the receiving (dispersing) terminal. Both of these terminals are collecting as well as dispersing, as is consistent with the line-based network architecture, but this case only studies one direction. Goods are collected by transporters serving terminal A. In the terminal, the goods are designated to a destination terminal (in this case, terminal B). A semi-trailer transports the goods to terminal B, where the goods are sorted once more and distributed to the consignees.
The following actors are identified in the case:

**Table 15 Actors and roles in case 1**

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unloader of goods</td>
<td>Unloads an incoming vehicle, such as a semi-trailer or lorry; places goods on a special area inside the terminal gate</td>
</tr>
<tr>
<td>Coordinator</td>
<td>Examines each waybill in and decides to which outgoing gate the goods are to be moved. The sorting system is based on the destination zip code.</td>
</tr>
<tr>
<td>Internal transporter</td>
<td>Moves goods from unloading area to the designated outgoing gate, often using a forklift</td>
</tr>
<tr>
<td>Loader of goods</td>
<td>Loads goods onto a semi-trailer or lorry at the outgoing gate</td>
</tr>
<tr>
<td>Driver of lorry</td>
<td>Operates a lorry (a self-powered transport resource)</td>
</tr>
<tr>
<td>Driver of semi-trailer tractor</td>
<td>Operates a semi-trailer tractor (used to transport unit loads, e.g. semi-trailers)</td>
</tr>
<tr>
<td>Loader of goods</td>
<td>Loads goods onto a semi-trailer or lorry at the outgoing gate</td>
</tr>
<tr>
<td>Driver of lorry</td>
<td>Operates a lorry (a self-powered transport resource)</td>
</tr>
<tr>
<td>Driver of semi-trailer tractor</td>
<td>Operates a semi-trailer tractor (used to transport unit loads, e.g. semi-trailers)</td>
</tr>
</tbody>
</table>
5.1.4 System description

The two terminals are very much alike. They work in the same way, look almost the same and are managed by the same principles.

The terminal consists of different organisational sections:

- **Goods reception**
  - Handles the reception of goods
  - Physical elements:
    - Gate
  - Loading area
- **Terminal flow coordination**
  - Decides destination gate for each consignment
  - Physical elements:
    - Check-in desk
- **Invoicing and finance**
  - Charges the customer with the price of their transport
  - Regulates financial compensation to hauliers
- **Sales office**
  - Receives orders from customers
  - Handles queries from customers
- **Transport planning**
  - Coordinates labour for hauliers
- **Terminal labour force**
  - Facilitates all movements of consignments within the terminal
  - Physical elements
    - Pallet trucks, manual
    - Pallet trucks, electric
- **Goods departure**
  - Handles departing goods
  - Physical elements
    - Gate
    - Loading area

**Goods reception**

The goods reception area consists of a gate, where a lorry or semi-trailer can dock, and a large square painted on the terminal floor next to the gate. When a vehicle arrives at the gate, the goods are unloaded and placed in the painted square. The lorry driver collects the waybills for all the consignments and proceeds to the check-in desk (containing the terminal flow coordination section). After the check-in process, the driver places each waybill on top of its consignment. The waybill is marked with the number of the destination gate.
Terminal flow coordination
The check-in desk is manned by a coordinator. When a consignment arrives at the terminal, this coordinator is handed the waybill (in three copies). He forwards one of the copies to the financial department. On first of the two remaining copies he writes the number of the departure gate that the consignment is to be transported to. The departure gates are numbered according to their destination terminals. The numbers are based on the first two digits of the destination terminal’s zip code. The marked waybills are returned to the consignment.

Invoicing and finance
Each waybill that arrives is passed through the check-in desk. One copy of each waybill is sent to the financial and invoicing desk. If the consignment has been delivered to consignee, a signed waybill confirms this and an invoice is generated. The financial desk also generates payments to the haulier that has transported the consignment.

Sales office
The sales office is the first – and in most cases the only – contact a customer has with the forwarding company. When a transport is ordered a sales representative creates a new entry in the order database (in a computer system). The order reception includes information such as:

- Address of the consignor
- Address of the consignee
- Invoicing address
- Type of goods
- Other terms or properties (such as time restrictions, whether or not the consignment contains dangerous goods etc.)

The sales office also handles inquiries about the state of existing consignments.

Transport planning
When an order is entered into the system, the transport planners can order a haulier to pick up the consignment from the consignor. Transport planning makes sure that enough semi-trailers are available for each destination by calculating the number of metres each consignment needs on a semi-trailer.

Terminal labour force
There are several trucks, electrical and manual, in the terminal. These are operated by terminal workers whose primary tasks include the transportation of consignments from the arrival gates to the destination gates that are indicated on their waybills.

5.1.5 Heterogeneous goods
This case does not contain any goods that are explicitly defined as heterogeneous by the forwarder except for consignments containing dangerous goods. Dangerous goods are treated in much the same way as normal goods, with a few deviations during loading and temporary storage.
5.1.6 Case chronology

The following sequence is typical for a consignment in the studied system.
1. The consignor prepares the goods for shipping (loads on pallet).
2. The customer orders transport of products. A waybill is created, either by the customer, or by the forwarder.
3. The forwarder relays a collecting order to a transporter.
4. The transporter sends a lorry to collect the consignment. The lorry collects several consignments of this type.
5. The lorry arrives at the collecting terminal with the load of collected goods, already on pallets. Each consignment has a waybill (three copies). Consignments containing dangerous goods also carry additional documentation.
6. The driver takes all of the waybills to the coordinator who takes one of the three copies and forwards it to the invoicing department. On one of the remaining two copies the coordinator writes the destination gate.
7. The driver returns to the vehicle with the two remaining copies of the waybills and unloads the goods.
8. The driver places the waybills (both copies) on each consignment.
9. An internal transporter reads the code for the destination gate on a consignment’s waybill.
10. The transporter uses a forklift to transport all pallets that are encompassed by the waybill to the indicated destination gate.
11. The driver arrives with a semi-trailer tractor.
12. The driver can choose to
   a) Remain until the semi-trailer is loaded or
   b) Leave the semi-trailer to be able to utilise the semi-trailer tractor and return when it is loaded.
13. The loader of the goods (often the driver) collects the waybills that are placed on the pallets at the destination gate.
14. The goods are loaded onto the semi-trailer. Dangerous goods are placed in the rear, near the opening (if the cargo is inspected by authorities; since the dangerous goods should be easy to find and verify).
15. The driver controls the loading, the cargo securing, and the documentation regarding dangerous goods.
16. The semi-trailer is sealed. The seal must remain unbroken up to the destination terminal.
17. The driver departs with the semi-trailer.
18. The semi-trailer from the collecting terminal arrives. As with the collecting terminal, the driver delivers the waybills to the coordinator.
19. The coordinator assigns each consignment to a destination gate. The destination gates are labelled with local zip codes for distribution.
20. The driver unloads the pallets and places the remaining copy of the waybill on each consignment (i.e. physically on the pallets).
21. A terminal worker, i.e. a person in the role of internal transporter, drives each pallet to the
gate assigned by the coordinator. The gate number is written on the waybills on the
pallets.

22. A lorry arrives. This lorry is not operated by the forwarder, i.e. not operated by the
terminal operator.

23. The driver of the lorry acts in the role of a loader of goods and drives the pallets onto the
lorry.

24. The driver of the lorry collects the waybills and takes them to the coordinator.

25. The coordinator receives one copy of each waybill, for registration purposes.

26. The lorry driver departs with the goods.

27. The consignment is delivered to the consignee.

5.1.7 Case conclusions

This case depicts the basic operation of a fairly standard line-based road network. The system
is based upon a high degree of standardisation. Physical standardisation is evident in the use
of pallets and semi-trailers throughout the system. These kinds of systems are often very
large, geographically, i.e. they consist of a large number of terminals covering a large
territory. Also, each terminal operates in exactly the same manner, enabling precise time
tables and schedules. The design of the system is evidently aiming at minimising the number
of direct interactions. The only time interaction is required, is when the physical properties of
the consignment exceed the boundaries of standardisation in the system. Dangerous goods are
noted and handled separately. Other exceptions are mainly triggered by divergent physical
dimensions such as a poorly loaded pallet.

5.2 Case 2 – Container exported to the Far East by road and sea

The forwarder makes plans for a container loaded with several consignments bound for the
Far East. The consignments come from different consignors and a firm sub-contracted to the
forwarder loads the container. The goods arrive at the sub-contracted firm that owns a small
terminal. Containers are filled, documented and transported to the port area. The forwarder
has booked a sea transport to the destination port.

5.2.1 Case objective

When more than one mode of transport is used, the interface between the consignment and its
environment is critical. This case is the first of several where intermodal transport is studied.
The objective was to find distinctions between how different goods types affected the
operation of the system.
5.2.2 Case methodology

Instead of physically following the goods as in Case 1, the use of information has been studied. The data comes from interviews with the forwarder (on one occasion coupled with follow-ups by telephone) and with the sub-contracted transport company (on one occasion coupled with follow-ups by telephone).

5.2.3 Actors and roles

The following actors are identified in the case:

*Table 16 Actors and roles in case 2*

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unloader of goods</td>
<td>Unloads arriving vehicles containing goods for container transport</td>
</tr>
<tr>
<td>Forwarder</td>
<td>Orders transport and filling of container from transporter</td>
</tr>
<tr>
<td>Transporter</td>
<td>Fills container and transports it to port area</td>
</tr>
<tr>
<td>Loader of goods</td>
<td>Loads container onto a vehicle/vessel, e.g. a ship</td>
</tr>
<tr>
<td>Driver of lorry</td>
<td>Drives goods (either to or from transporter terminal)</td>
</tr>
</tbody>
</table>

5.2.4 System description

This system is built around the ISO-container as a transport unit. The transport chain is described in Figure 43 below.

*Figure 43 Case 2 consists of a transport between Sweden and the Far East. The two terminals A and B are connected via a sea transport between the ports P1 and P2. The sea transport is most probably a multi-leg operation with several transhipments.*

The system is built around a trans-continental sea voyage. All deadlines, delivery times and planning efforts serve to support the schedule of the shipping line. The shipping line itself is not involved in the transportation of goods to and from the end customers, but only in the
The port-to-port part of the flow. The Port-to-Door and Door-to-Port sections of the flow are handled by the shipping line’s customers – the forwarding companies. Each forwarding company has their own routines and regulations, but they are in general operating the way that is described in this case. There is a notably high amount of documents involved in transporting a consignment overseas by ship. Several of these are produced without the consignor and consignee knowing/being aware of this. This leads to a complicated information structure. Almost all the actors are responsible for the production or interpretation of at least one new document. Most of these documents are to be read and understood by operatives anywhere on the transport chain.

As stated earlier, the main focus in this study is that of the information structure. The following documents are used in the system:

- **Collection waybill, CMR**
  - Used for the road transport to the transporter terminal
  - Produced by the consignor or by the transport company delivering the consignment to the transporter terminal
- **Container manifest**
  - Contains a list of the contents of a container.
  - Produced by the forwarder
- **Dangerous Goods Declaration – DGD**
  - A document stipulating the nature of the dangerous goods as regulated by the International Maritime Organization (IMO). This document is produced by the consignor.
- **Port of Singapore Authority (PSA) documentation**
  - If the Port of Singapore is to be used, stricter rules apply regarding the documentation of dangerous goods.
  - These documents are often produced by the forwarder or by the consignor on the forwarder’s request. Many consignors are not aware of the transport route and may therefore inadvertently cause a shipment to be rejected in Singapore because of these rules.
- **Container Packing Certificate – CPC**
  - The operator packing the container signs a document where he declares:
    - The types and quantities of dangerous goods loaded in the container
    - That several regulations and stipulations have been followed
- **Container waybill**
  - A document accompanying the container throughout the transport chain
- **Customs declarations**
• Bill of Lading – BoL
  o The forwarder composes this document that works as a requisition order for the consignee
  o The BoL is sent by mail to the consignor
  o At the discretion of the consignor, the BoL is mailed to the consignee. The consignee cannot retrieve the consignment without this document.

• Shipping Advice
  o This document contains information on what consignments are on a ship and when it will arrive at the destination port.
  o After the ship has sailed, the forwarder sends the Shipping Advice by fax to his counterpart on the receiving end.

• Invoices
  o Several of the companies in this system produce invoices to, as well as receive them from, each other.

The transport itself takes about a month, giving the actors plenty of time to produce and process most of the information.

5.2.5 Heterogeneous goods

Every consignment in this case can in a way be defined as heterogeneous since no two transport operations are alike. However, a significant difference is found when transporting dangerous goods in terms of documentation and need for control. Therefore, for this system, the dangerous goods are considered heterogeneous.

5.2.6 Case chronology

The events in this case take place in the following sequence:

1. The consignor prepares the goods for shipping.
2. The customer orders a transport of products. A waybill (CMR) is created by the forwarder based on the following information:
   a) Type of goods
   b) Quantity
   c) Type of packaging
   d) Dangerous Goods classification and information
   e) Destination
   f) Pickup time
   g) Sailing day
3. If the consignment contains dangerous goods, the customer faxes a document called Dangerous Goods Declaration (DGD) containing specific information on the goods. This document is required for sea transport.
4. The DGD is forwarded to one of two actors:
   a) The shipping agent. This is the case if the forwarder has enough goods to fill a container.
   b) A cooperating forwarder. If a container cannot be filled, the forwarder arranges for cooperation with another forwarding agent, so that the container will be full. The cooperating forwarder fills the container and takes care of the documentation.

5. When the shipping agent receives the booking, the documentation including any DGD’s is forwarded to the shipping line headquarters.

6. The goods are transported to a terminal operated by the transporting agent.

7. The transporting agent fills the container, secures the cargo and issues a CPC.

8. The transporting agent transports the container to the port area.

9. Stevedores load the container onto a feeder vessel.

10. The feeder is unloaded on the continent and the container is reloaded on a trans-oceanic vessel.

11. When the container arrives at the destination port, it is transported to a representative of the forwarding company.

12. The container is opened and the various consignments are delivered to their consignees.

Interactions between consignment and the TCS are both easy and difficult to ascertain in this case. It is easy since the consignment actually ceases to exist once it is embedded within the container, thereby allowing it to be a “passenger” of the container. There is true interaction only in the terminals where the container is packed or opened. However, even though the consignment is encapsulated in the container, some of its properties may still be visible. If the consignment contains dangerous goods, the container is labelled accordingly. If the consignment is heat sensitive, the container has a cooling system. Therefore, some interaction will take place between the container and the different TCSs along the route. This may turn problematic since the container often carries several consignments bound for different consignees and sent by different consignors. An interaction resulting in a state change may be beneficial to one consignment but may be detrimental to another if they are packed into the same container.

5.2.7 Case conclusions

In this case, no two transport operations are alike. The only things that are standardised are the ISO-container and the documentation. A transport like the one described here has several interested parties. Each container may contain several consignments. Each of these consignments has different consignors and consignees. The roles of transport initiators may differ between the consignments; some are initiated by the consignors, some are not. As long as the transport is executed according to plan, the standardised interfaces of the container and the documentation are adequate to ensure that the container achieves its goal state. However, events may occur where these interfaces are not sufficient. It is in these cases that the forwarder may be faced with a decisively larger amount of non-standardised interfaces. If, for instance, consolidation is broken, the container yields a new interface for each of the contained consignments. These interfaces were previously encapsulated within the container but are suddenly presented and must be managed. The interface that the forwarder can utilise...
to control individual consignments may also be hampered by for instance geographical distance, language problems and cultural differences.

5.3 Case 3 – Chemicals in a tank container transported by RoRo ferry, road and rail

The transport chain, of which this container is a part, is a so-called “dedicated flow” meaning that the individual containers travel back and forth between the same consignee and consignor repeatedly. The studied part of the chain begins when the container arrives by ferry and is transported by road to the intermodal terminal. A rail transport takes the goods to the destination terminal where they are transported by road to the consignee. The container is emptied, cleaned at a nearby cleaning facility and returned by the same route to the consignor. In this system, the occurrence of dangerous goods is very common, and the forwarder has therefore decided that all goods types should be equally treated, dangerous or not.

5.3.1 Case objective

The objective of this case was to study whether repetition and routine had any impact on the intensity of the interaction between consignment and the role of the forwarder.

5.3.2 Case methodology

As with Case 2, this case is based on interviews. The goods have been followed from the port area to the consignee and several of the actors in between have been interviewed first-hand. In addition to the interviews, several observations were also made that contributed to the overall picture of the case.
5.3.3 Actors and roles

The following actors are identified in this case:

Table 17 Actors and roles in case 3

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>International forwarder</td>
<td>Sends containers to Swedish consignees from international consignors</td>
</tr>
<tr>
<td>Swedish forwarder</td>
<td>Orchestrates entire transport on Swedish soil</td>
</tr>
<tr>
<td>Port operator</td>
<td>Operates a ferry port in Sweden and controls all goods going in and out of the port area</td>
</tr>
<tr>
<td>Ferry operator</td>
<td>Operates a ferry line between two countries and controls the goods on the ferry. The ferries are RoRo with capacity for rail wagons.</td>
</tr>
<tr>
<td>Stevedores</td>
<td>Load and unload the ferries</td>
</tr>
<tr>
<td>Local haulier 1</td>
<td>Collects goods at terminal (port or intermodal terminal) and hauls locally, often to another terminal.</td>
</tr>
<tr>
<td>Local forwarder</td>
<td>Acts as TCS for the last leg of the transport from the intermodal rail terminal to the consignee, and then via cleaning depot back to the intermodal rail terminal</td>
</tr>
<tr>
<td>Local haulier 2</td>
<td>A subcontractor to the local forwarder that performs the actual transport</td>
</tr>
<tr>
<td>Intermodal terminal operator</td>
<td>Operates an intermodal terminal. Loads and unloads rail wagons.</td>
</tr>
<tr>
<td>Cleaning depot</td>
<td>Cleans the emptied tank after unloading</td>
</tr>
<tr>
<td>Consignee</td>
<td>Receives the container</td>
</tr>
</tbody>
</table>

5.3.4 System description

This system is based on routine. The forwarder manages the import of tank containers transported from a chemical industry in Germany to another chemical industry in the north of Sweden. The transport is multi-modal, utilising both road, rail and ferry transport.
Figure 44 The case consists of a transport of one tank container. A road transport is followed by a ferry and a short road leg before the rail portion. The last leg (the delivery to the consignee) is by road. The ferry is the same as in Cases 4 and 5.

The containers are dedicated, i.e. they move back and forth in the same consignor-consignee relation, creating a steady flow of goods conceptually resembling a discrete “pipeline” between the companies. The pipeline analogy is enhanced further as the system is studied more deeply. All of the actors in the chain are familiar with this specific relation and how they are expected to perform. There is seldom any variation in the flow, and when they occur there are clear routines on how to handle them. All disturbances are reported to the forwarder, often by fax or phone. The forwarder also issues most of the documents in the system and is therefore directly responsible for the quality of the information.

5.3.5 Heterogeneous goods

Since there is little or no physical difference between two products – they are loaded in tank containers – the system handles differences in goods type extremely well. One of the more distinguishable differences is the occurrence of dangerous goods. Depending on dangerous goods classification, a container may be subject to handling and loading restrictions. These restrictions may also vary between the different modes of transport. In the chemical industry, the occurrence of dangerous goods is common. In the studied flow, 80% of the tanks contained dangerous goods. Non-dangerous goods were the minority, which reflected in the routines. Every transport should contain dangerous goods documentation as a rule, with the exception of the 20% of non-dangerous goods where the documents were to be left out.
5.3.6 Case chronology

1. The international forwarder sends a fax to the Swedish forwarder containing:
   a. Estimated time of arrival (ETA) of ferry with consignment on board
   b. Name of consignor
   c. Name of consignee
   d. Name of product, dangerous goods class and UN number
   e. Name of consignor
   f. Information on whether the tank container is to be returned to the consignor.

2. The Swedish forwarder notifies the consignee about the coming delivery (by phone).

3. The Swedish forwarder notifies the intermodal terminal operator about the consignment. This notification is sent by fax and acts as a reservation.

4. The intermodal terminal operator registers the order in their computer system as a reservation on the specific train.

5. The intermodal terminal operator marks the fax with the date and returns it to the forwarder as confirmation.

6. The Swedish forwarder notifies the ferry operator by fax about the consignment.

7. The ferry operator returns the fax as confirmation.

8. If the tank container is to be returned, the Swedish forwarder books this transport by fax to the local forwarder, the ferry line, and to the intermodal terminal operator.

9. A CMR is created by the consignor for the transport to the Swedish intermodal terminal.

10. The tank container is loaded by the international forwarder. The international forwarder confirms that loading is finished by re-sending the original order fax to the Swedish forwarder, only this time with loaded quantity added.

11. The Swedish forwarder produces several documents:
   a. Waybill for the final transport from the intermodal terminal to the consignee (1 original and 3 copies)
   b. One DGD for “Emptied, not cleaned” tank container
   c. One DGD for “Emptied, cleaned” tank container
   d. Dangerous goods Sender certificate
   e. Copies a DG fact sheet for road transport if it is available as a precaution since it is not always present in the document tube on the container.
12. The Waybill is stamped with a checklist for the Local haulier 2 or the Local forwarder to fill
   a. Original + Copy 1 is stamped with a checklist containing:
      i. Lorry ID
      ii. Delivery time
      iii. Goods temperature (if needed)
      iv. Name and signature of driver (acts as a proof of delivery)
   b. Copy 3 is kept by the Swedish forwarder for future reference
13. The above documents are sent by mail to the Local haulier 2
14. The Swedish forwarder faxes a copy of the Waybill to the Local forwarder as notification
15. Stevedores load the tank onto a temporary semi-trailer owned by the shipping line. The port authority checks the documentation for compliance with dangerous goods regulations. As the ferry is loaded, information regarding the container is entered into the shipping line’s computer system. This document is called the manifest and contains information on all cargo on board the ferry. It is completed half an hour after departure
16. The ferry departs
17. The customer of the shipping line (in this case the Swedish forwarder) receives a fax sent by the shipping line’s computer system. The fax contains the same information this is in the manifest for this specific cargo (container number, weight etc.).
18. The Ferry operator faxes to the Local haulier 1 a list of the containers needing transport to the intermodal terminal
19. The Intermodal terminal operator calls the Ferry operator to ensure that the container is on the ferry
20. The Intermodal terminal operator plans exactly how the train will be loaded, mainly regarding restrictions due to dangerous goods etc. A rail wagon containing dangerous goods may not be placed first, last or next to another wagon containing dangerous goods.
21. The Intermodal terminal operator informs the handling personnel about the train configuration
22. At arrival, the ferry is unloaded by Stevedores. All documents regarding the individual container are handed over to the ferry operator. The trailers are moved by the Stevedores to a dedicated area within the port.
23. The Local haulier 1 sends the first semi-trailer tractor to collect all the freight documents (except documents regarding dangerous goods as that would be illegal) from the Ferry operator to deliver with the first trailer to the Intermodal terminal operator
24. The Local haulier 1 transports the semi-trailer to the intermodal terminal
25. If the Local haulier 1 is early, the tank is temporarily placed within the terminal perimeter. Dangerous goods have a special area.
26. The tank is loaded onto a train by the handling personnel. Mostly the containers are lifted from the trailer directly onto the train. The containers that have been temporarily placed on the ground are loaded onto the train.

27. The train is transported to the destination terminal.

28. The train is unloaded by the Intermodal terminal operator in the destination terminal.

29. The Local haulier 2 transports the container to the Consignee and empties it. The waybill is signed as a proof of delivery. The Consignee keeps Copy 2.

30. The Local haulier 2 cleans the tank at a cleaning facility and receives a cleaning certificate.

31. The Local haulier 2 returns the tank to the intermodal terminal for return trip back to consignor.

Direct interaction between the individual consignment and the TCS is relatively scarce. Because of the standardised interface – i.e. the use of an ISO-container – the actors taking on the TCS role throughout the system rarely need to know what is in the consignment or any other specific properties. Direct interaction becomes necessary at some stages when the tank contains dangerous goods, but apart from that only the end points are interacting with the consignment.

5.3.7 Case conclusions

In this system, 80% of all consignments are classified as dangerous goods. This, plus the fact that the container is dedicated contributes to a high degree of encapsulation of data. Even though there are several attributes on the documentation that accompany a transport, few of them vary between occasions. Thus, the standardisation does not only reduce the width of the interfaces, but also the depth (in some instances to a high degree). Every transport in this chain is executed in the same manner by the forwarder, regardless of dangerous goods classification. This means that the 20% of the consignments that are not classed as dangerous goods are treated in almost the same way. The only difference is that two documents are not included in the information exchange.

5.4 Case 4 – Road and RoRo ferry transport of semi-trailer

A semi-trailer is exported by ferry and later loaded onto a rail wagon for transport to its destination terminal, where it is transported by road to the consignee. The studied system does not contain dangerous goods but experiences problems due to poor cargo securing. The goods type is large rolls (not on pallets). The ferry used is the same as in Cases 3 and 5.
5.4.1 Case objective

Case studies 4 and 5 were performed simultaneously with the purpose of comparing two similar supply chains. Both chains use rail transport in Germany. Both chains also use the same ferry link to transport semi-trailers from Sweden to Germany. In Case 4, the semi-trailer is transhipped to rail in Germany and in Case 5, rail is used for the long haul leg in Sweden. The comparison is focused on the intensity and frequency of interactions between consignment and actor assuming role of a TCS.

5.4.2 Case methodology

The case study is based on interviews with the various information handlers in the supply chain. First-hand observations have also been made, and copies of documents have been kept for future reference. Since the study was made simultaneously with case 5, the ferry operator was interviewed about both cases at the same time.
5.4.3 **Actors and roles**

The following actors were identified:

*Table 18 Actors and roles in case 4*

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consignor</td>
<td>Manufacturing company in the south-eastern part of Sweden. The company is multi-national.</td>
</tr>
<tr>
<td>Consignee</td>
<td>Belongs to the same corporation as the consignor. Located in Germany and acts as a distribution hub with warehousing facilities.</td>
</tr>
<tr>
<td>Main forwarder</td>
<td>Situated in the southwest of Sweden. Controls the entire flow including booking of hauliers and ferry transport (in Sweden).</td>
</tr>
<tr>
<td>Swedish sub-forwarder</td>
<td>Local branch of the main forwarder, responsible for the consignor contact. Situated in the same town as the consignor.</td>
</tr>
<tr>
<td>Road hauliers</td>
<td>Small (most often) transport company tied to the main forwarder’s network. The road haulier performs two tasks: delivers an empty semi-trailer to the consignor and transports the loaded semi-trailer from the consignor to the port.</td>
</tr>
<tr>
<td>Ferry operator</td>
<td>Runs a ferry line between the southwest of Sweden and Germany. The ferry is equipped for RoRo as well as rail wagons.</td>
</tr>
<tr>
<td>Trailer-auditing company</td>
<td>The ferry operator has outsourced the auditing of all semi-trailers arriving at the port area to the auditing company</td>
</tr>
<tr>
<td>Swedish stevedores</td>
<td>Company of stevedores performing the loading/unloading of the ferries.</td>
</tr>
<tr>
<td>German stevedores</td>
<td>Company of stevedores performing the loading/unloading of the ferries.</td>
</tr>
<tr>
<td>German sub-forwarder</td>
<td>German branch of the Swedish forwarding company (all forwarders in this case belong to the same corporation).</td>
</tr>
<tr>
<td>German rail terminal operator</td>
<td>Operates a rail terminal where trains are loaded and unloaded.</td>
</tr>
<tr>
<td>German intermodal operator</td>
<td>Operates a network of rail terminals and offers transport between these.</td>
</tr>
<tr>
<td>German road haulier</td>
<td>Transports semi-trailer from rail terminal to consignee.</td>
</tr>
</tbody>
</table>
5.4.4 System description

This case describes a regular transport of finished products from a factory in Sweden to a distribution hub in Germany. The factory manufactures carpets that take on the form of large rolls when transported. The rolls are too big for standard EUR-pallets (they are 2, 3, or 4 meters long), and cannot for this reason be intermixed with palleted goods in a standardised line-based terminal network as described in case 1.

![Diagram](image)

*Figure 45 This case consists of a semi-trailer being transported by ferry (between ports P1 and P2) and then put on a train (R1 to R2). The final stage is by road. The ferry is the same as in cases 3 and 5.*

There is an average of three semi-trailers per day travelling the studied route. The studied company has chosen to use an intermodal transport of the semi-trailers in Germany because of the German regulations, stipulating an allowance of higher weights for trailers transported by train, if the intermodal terminal at the destination is less than 100 km from the final destination.

5.4.5 Heterogeneous goods

The consignor has a problem with the securing of the goods in the semi-trailer. About 10% of the semi-trailers that travel the same route each week display – or run a risk of – load shifting according to the German intermodal operator. This carries a considerable extra cost for the consignor since the load often has to be secured again, once the ferry unloads in Germany. Coupled with the fact that the physical measurements exceed the pallet, these consignments are all heterogeneous.

5.4.6 Case chronology

1. The consignee places an order with the consignor. The order is placed in a joint computer system. Since both companies belong to the same corporation, they have online contact through the same warehousing/ordering system.

2. The consignor phones the Swedish sub-forwarder and delivers information about the date, approximate weight and destination of the consignment. Since this flow is highly frequent, the consignor does not need a confirmation from the forwarder. If something goes wrong, the forwarder will contact the consignor.
3. The Swedish sub-forwarder makes a booking in their computer system, which is shared by all offices, connected to the forwarder. In the computer system, all the information later printed on a CMR (an international waybill) is listed (as well as some internal information directed at the forwarder only, such as invoicing information etc.). The order is electronically promoted to the main forwarder through the computer system.

4. The main forwarder’s planning centre directs the vehicle with the most favourable position (and with an empty semi-trailer) to the consignor to pick up goods as well as leave the empty semi-trailer in exchange. This ordering is done by mobile phone. The planning centre gets the semi-trailer number of the empty semi-trailer from the driver. This number is relayed to the Swedish sub-forwarder by e-mail, fax or phone (whichever is convenient at the time). At the same time, the main forwarder books the ferry by fax to the ferry operator. The booking of the ferry is done once a day, by batch, since the main forwarder is a large customer to the ferry operator and books several semi-trailers for each departure. The ferry operator enters the bookings in their own computer system. The ferry operator checks a box on the forwarder’s booking fax and returns it (by fax) as a confirmation.

5. The consignor produces a loading list containing order numbers and the weight for each order along with the semi-trailer number. This list in combination with the packing lists, containing each of the orders in detail, enables the warehousing personnel to load an empty semi-trailer standing by at the consignor. The forwarder always keeps 3-4 empty semi-trailers standing by for immediate use by the consignor.

6. The consignor produces a waybill accompanying the goods and aimed at the consignee. The document contains addresses to consignor and consignee, the semi-trailer number (not the licence plate number) and the gross weight of the goods. This waybill is faxed to the Swedish sub-forwarder.

7. The Swedish sub-forwarder then constructs a CMR (an international waybill) from the information already in the computer system updated with the actual weight of the consignment, the semi-trailer number etc. from the faxed waybill. Copies of the CMR are data-faxed to the consignee and the German sub-forwarder as notification. The original of the CMR is put in the semi-trailer pouch, which is picked up by the truck driver as he collects the semi-trailer.

8. The semi-trailer is driven to the port area where it is placed within the gates of the port.

9. The driver signs a semi-trailer declaration where he confirms the information on the CMR as well as ensures that the requirements concerning sea transport of dangerous goods are met. This semi-trailer declaration also serves as a security check since the driver is not permitted to drive further into the port area until his declaration has been approved. At the same time a trailer-auditing company working for the ferry operator examines the semi-trailer. The purpose of this audit is to ensure the seaworthiness of the semi-trailer (with regard to load shifting etc). The driver surrenders the trailer pouch to the ferry operator.

10. The ferry operator updates the registration of the semi-trailer in their computer system. These registrations are the basis for the manifest (a list of the various load units travelling on the ferry) which is preliminary until the loading of the ferry is finished. The auditing personnel have close contact with the ferry operator and tell the driver where to put the semi-trailer. This location is decided upon in collaboration with the ferry operator and the stevedores, and its purpose is to facilitate the loading procedure once the ferry arrives.

11. Copies of the manifest are held by the captain of the ferry (or his representative) and the stevedore company. The stevedores load the ferry, having close radio contact with the ferry operator.
12. The ferry leaves Sweden for Germany.

13. All forwarders that have booked transport on the ferry are automatically data-faxed with confirmation whether their semi-trailers were loaded onto the ferry or not and in some cases they are not. This can be caused by a late arrival, an audit failure or some other mechanical error (flat tyre etc.). In these cases the ferry operator contacts the forwarders and the semi-trailer is rescheduled. At the same time, the manifest is sent through the ferry operator’s own computer system to their office in Germany. The ferry operator has chosen to work in this manner that each office is regarded as a separate unit and that the information is sent at the sender’s discretion (not shared online). The system enables online access by both offices, and although this feature is rarely used, it sometimes is necessary for the Swedish office to look up an order received by the German side or vice versa.

14. The German stevedores unload the ferry and put the semi-trailers going by rail in a certain terminal. They also perform an audit as to the semi-trailer’s ability to safely go by German rail. Sometimes, load shifting occurs during the sea transport causing the stevedores to correct the load. In this case study approximately 10% of all semi-trailers, originating from this particular consignor, needs correction. Typically, this costs around €30 each time.

15. The semi-trailer is then transported by rail and later by road to the consignor.

5.4.7 Case conclusions

For a highly repetitive flow, the number of interactions that are necessary between forwarding actor and consignment is high. There are high degrees of manual work, especially when changing transport modes. The most costly of all interactions in this case is the transhipment in Germany between road and rail. 10% of the consignments experience load shifting during the sea transport. The German rail operator does not accept this and corrects the shift. This costs about €30 each time.

Delays are not a concern in this case, since the consignee has chosen to co-locate the distribution hub with a warehouse. According to the Swedish forwarder, the consignee maintains a high stock level and is therefore insensitive to delays.

5.5 Case 5 – Road, rail and rail ferry transport of semi-trailer

Case 5 is similar to Cases 3 and 4, but involves a semi-trailer loaded on a rail wagon and then transported on a rail ferry (the same as in the previous two cases). The company and transport chain is different from Case 4, but similar in one aspect; the goods type is large rolls. Rolls are often too big to be put on standard pallets and handling them is difficult.
5.5.1 Case objective

Case studies 4 and 5 were performed simultaneously with the purpose of comparing two similar supply chains. Both chains use rail transport in Germany. Both chains also use the same ferry link to transport semi-trailers from Sweden to Germany. In Case 4, the semi-trailer is transhipped to rail in Germany and in Case 5, rail is used for the long haul leg in Sweden. The comparison is focused on the intensity and frequency of interactions between consignment and actor assuming role of a TCS.

5.5.2 Case methodology

The case study is based on interviews of the various information handlers in the supply chain. First-hand observations have also been made, and copies of documents have been kept for future reference. Since the study was made simultaneously with case 4, the ferry operator was interviewed for both cases at the same time.
5.5.3 Actors and roles

The following actors were identified:

Table 19 Actors and roles in case 5

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consignor</td>
<td>Manufacturing company in the northern central part of Sweden. The company is multi-national and has a subsidiary in Germany.</td>
</tr>
<tr>
<td>Consignee</td>
<td>Customer of the consignor. Situated in Germany.</td>
</tr>
<tr>
<td>Main forwarder</td>
<td>Situated in the Stockholm area of Sweden. Controls the entire flow including bookings of hauliers and ferry transport (in Sweden).</td>
</tr>
<tr>
<td>Swedish road haulier</td>
<td>The road haulier performs two tasks: delivers an empty semi-trailer to the consignor and transports the loaded semi-trailer from the consignor to the rail terminal. Owned by the Swedish rail operator.</td>
</tr>
<tr>
<td>Ferry operator</td>
<td>Runs a ferry line between the southwest of Sweden and Germany. The ferry is equipped for RoRo as well as rail wagons.</td>
</tr>
<tr>
<td>German sub-forwarder</td>
<td>German partner of the Swedish forwarding company.</td>
</tr>
<tr>
<td>Intermodal operator</td>
<td>Swedish company specialising in intermodal solutions. Has responsibility for the entire rail transport, including loading/unloading of rail wagons.</td>
</tr>
<tr>
<td>Swedish rail terminal operator’s central booking office</td>
<td>Centralised order handling for all the intermodal rail terminals in Sweden.</td>
</tr>
<tr>
<td>Swedish rail terminal operator</td>
<td>Lifts semi-trailers onto trains and transports them to another rail terminal for unloading/reloading. Owns the road haulier.</td>
</tr>
<tr>
<td>German intermodal operator</td>
<td>Arranges the intermodal transport in Germany</td>
</tr>
<tr>
<td>German rail terminal operator</td>
<td>Operates a rail terminal where trains are loaded and unloaded.</td>
</tr>
<tr>
<td>German road haulier</td>
<td>Transports semi-trailer from rail terminal to consignee.</td>
</tr>
</tbody>
</table>

5.5.4 System description

In this case, a paper mill produces large paper rolls. These are sent from Sweden to Germany loaded on a semi-trailer that is transported from Sweden by rail and rail/ferry. In the studied case, the paper company in Sweden has a subsidiary in Germany that in turn has a customer (the consignee). The subsidiary puts an order from the consignee into a joint computer system. The consignment is prepared and shipped from Sweden by trailer, loaded onto a rail
wagon in an intermodal terminal and transported on a ferry for subsequent rail transport to the
closest intermodal terminal to the consignee. It is transhipped and transported by road the last
leg.

Figure 46 A semi-trailer is put on a rail wagon (at R1) and ferried on a rail ferry (between P1 and P2).
The latter part of the transport is similar to the corresponding part in case 4 (P2 to R2). The ferry is the
same as in Cases 3 and 4.

In this case, the specific consignment studied was late for the ferry departure, thus arriving
one day late in Germany. The German sub-forwarder faxed to the main forwarder who also
got a deviation report from the intermodal operator. The main forwarder notified the
consignor who, in turn, notified the consignee about the delay.

5.5.5 Heterogeneous goods
This type of transport is fairly frequent (around 5 trailers per day). In this system, according to
the consignor, no goods are considered heterogeneous. The fact that the goods consist of large
rolls is not an issue here, since that is the norm for the paper company.

5.5.6 Case chronology
1. The consignee places an order from the consignor. The order is placed in a joint computer
system. In this computer system, the consignee can by himself register orders, which are
detected by the consignor and acted upon. The computer system belongs to the consignor
and is used throughout their multi-national corporation.

2. 1 to 3 days before the scheduled departure a “Loading list” is generated from the computer
system. The document contains information regarding the destination, the approximate
weight and the departure day. This list is faxed to the main forwarder. The main forwarder
registers the upcoming transport in their computer system, which is an internal system,
shared by the main forwarder’s offices in Sweden. This system in the end generates the
invoice to be sent to the consignor.

3. The main forwarder calls the haulier and tells him which of the semi-trailers to deliver
empty to the consignor for the loading. 3-4 empty semi-trailers are always standing by at
the closest rail terminal (30-40 km away).

4. The main forwarder calls the consignor and gives him the semi-trailer number (not the
registration number). At this time he implicitly confirms the order.
5. 1 day before departure, the main forwarder calls the intermodal operator and gives him the desired ferry departure. The order is registered in the Swedish intermodal operator’s computer system, sometimes lacking semi-trailer number, weight and rail wagon number. The main forwarder and the intermodal operator have frequent phone contact so that possible misunderstandings etc. are avoided. The missing information is delivered at a later stage from the main forwarder or the rail terminal operator via phone. Even though the Swedish intermodal operator has the ability to receive bookings through their web page, the main forwarder always phones in his orders. The reason for this is that he does not have Internet access. The intermodal operator data-faxes the Swedish rail terminal operator’s central booking office and reserves a rail wagon for the specific date. This information is also (in most cases) faxed to the rail terminal. The intermodal operator calls the ferry operator and reserves a certain amount of metres on the desired ferry.

6. The specific semi-trailer is hauled to the consignor on the appointed day. The driver states a reference number from the loading list created in stage 2 as identification.

7. The consignor loads the semi-trailer, although the driver is responsible for securing the cargo. The warehousing personnel have picking orders and packing lists generated from the order system to their aid, when loading the semi-trailer.

8. During the loading of the semi-trailer, a document called “Internationaler Frachtbrief/International consignment note” is produced. One copy is faxed to the main forwarder and one accompanies the goods to the consignee. To date, another copy is produced and sent with the goods, for customs, and although this copy is not needed anymore (became unnecessary when Sweden joined the EEC) it is still printed. The consignor also produces a document called “Weightspecification/packing list” for the consignee to verify the goods with.

9. The main forwarder’s computer system generates a notification, which is data-faxed to the German sub-forwarder.

10. The haulier puts all the documents bound for the consignee in a trailer pouch attached to the semi-trailer before departing for the rail terminal.

11. At the rail terminal the semi-trailer is lifted onto one of the Swedish intermodal operator’s rail wagon. There are always a number of wagons standing by at the rail terminal belonging to the Swedish intermodal operator. The terminal operators have a list where the trailer number is connected to a rail wagon number as well as the destination (which rail terminal) of the goods.

12. The rail terminal operator has a planning sheet where all relevant information regarding the individual wagons as well as the complete trains are noted. This document is faxed to the Swedish rail terminal operator’s central booking office. The document is completed with the order number (generated when the transport was booked as described under step 5) and returned to the rail terminal operator by fax. The rail terminal operator confirms that the train is loaded by faxing the complete planning sheet once more to the Swedish rail terminal operator’s central booking office. The Swedish rail terminal operator’s central booking office orders an engine to transport the train assembled at the rail terminal to the ferry.

13. The intermodal operator calls the rail terminal operator and is provided with the rail wagon number. This is entered into the intermodal operator’s computer system.

14. The intermodal operator data-faxes a notification, of all their wagons in the planned train, to the ferry operator, the German intermodal operator and to the Swedish rail terminal operator’s central booking office.
15. The rail wagon is then transported directly to the ferry terminal (by the engine ordered in item 12) and loaded onto the ferry for transport to Germany. The ferry operator does not in any way interfere with the loading process and no information is passed between rail operator and ferry operator. The rail operator, not the stevedores, conducts the loading.

16. In Germany, the rail wagon is transported to the destination terminal, where the semi-trailer is unloaded.

17. The semi-trailer is hauled to the consignee.

18. If the semi-trailer for any reason is delayed or missing, the intermodal operator faxes a deviation report to the main forwarder, who informs the consignor. In this case, the delay of the semi-trailer was due to late arrival in Germany by the ferry, thus causing the goods to miss the train (causing a 24-h delay). The German sub-forwarder noted that the semi-trailer was missing and returned the notification fax sent by the main forwarder (see item 9 in this list).

5.5.7 Case conclusions

The number of interactions is fairly high also in this case. Since both Cases 4 and 5 describe high-frequency transport, where the routine level is extremely high, the number of direct interactions between the individual consignment and its TCS should by all accounts be kept to a minimum. These systems (Cases 4 and 5) give the impression of having evolved from small-scale solutions, which have become more frequent with time.

All the involved actors “know” one another professionally and they always resort to the phone when something goes wrong. These systems are both examples of the term “practice makes perfect”, because not one of the pieces of information used is without purpose and although the means of transmitting information leaves more to be desired, the content of the different documents is exactly what is required.

5.6 Case 6 – Internal handling in warehouse

A large paint and adhesives manufacturer has, in the same facility as one of its factories, a warehouse functioning as a national hub. The goods stored in the warehouse arrive via semi-trailer or from the internal production. When a pallet is stored in the warehouse, a card index is used to file its position.

5.6.1 Case objective

This case study was initially focused on the pre-consignment phase, i.e. the consignor’s preparations for a consignment. The case has since then evolved and encompasses several “mini-transports” where consignors, consignees and forwarding actors are operators from the same company, working in the same warehouse. The case focuses on the consignment-forwarder interactions on a micro-level where each interaction is between a person and a product or a pallet. The case illustrates the potential complexity and diversity of a seemingly
simple and straightforward operation that is common practice throughout the manufacturing industry.

5.6.2 Case methodology

The majority of the data in this case was collected in interviews with the staff in the studied warehouse. The interviews were carried out on three separate occasions during a 5-month period. The interviewees were confronted with the data and were given opportunity to revise earlier statements. On two of the three occasions, additional researchers were present during the interviews (a professor and his PhD student from another university as well as a fellow PhD student working on the same project). All notes were compared afterwards.

5.6.3 Actors and roles

This case differs from the others since it does not encompass any transportation- or forwarding companies but rather a warehouse operation. The transports in the studied system were carried out by various forklifts (manual, powered, and automatic), Automated Guided Vehicles (AGV) and a conveyor belt. The role as TCS is assumed by the card file station that is situated in the centre of the warehouse. The card file station acts as a hub for all incoming shipments, either from the production system in the factory or from external consignors. Two actors/roles have been identified in this case:

Table 20 Actors and roles in case 6

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card file operator</td>
<td>The TCS of the system</td>
</tr>
<tr>
<td></td>
<td>Designates warehouse slots for all incoming goods</td>
</tr>
<tr>
<td></td>
<td>Initiates fetching of goods for shipment to customers</td>
</tr>
<tr>
<td></td>
<td>Operates the high stacker forklift</td>
</tr>
<tr>
<td>Forklift operator</td>
<td>Transfers goods from the conveyor belt to designated slots in the warehouse</td>
</tr>
<tr>
<td></td>
<td>Retrieves goods from the warehouse and prepares them for shipment to customers</td>
</tr>
<tr>
<td></td>
<td>Picks orders that are less than one unit load, i.e. consists of several products that are consolidated onto one pallet.</td>
</tr>
</tbody>
</table>

5.6.4 System description

The studied warehouse has two gates where goods arrive and two gates where goods depart. One of the incoming gates is trafficked by an AGV-train from the adjacent factory. The train loads four pallets and arrives every ten minutes. The train automatically unloads the pallets onto the conveyor belt so that the card file station can designate each pallet to a slot in the warehouse.
When loading the AGV-train, the personnel in the factory produce an internal waybill, a document in two copies containing the following information:

- Product name
- Product number
- Manufacturing batch number
- Manufacturing date
- Packaging size (there are several different sizes, consolidated into boxes)
- Number of boxes/packages

This waybill is placed on the pallet and is later used by the card file station when filing the warehouse position of the pallet.

The studied warehouse also acts as the company’s national hub. This means that products also arrive from other factories and are unloaded through the “incoming goods” gate. The consignments are unloaded onto the conveyor belt. No internal waybill exists for these consignments, since they are not produced in the adjacent factory. The card file operator therefore has to produce a document fulfilling the same purpose as the internal waybill.

The terminal has two outgoing gates. The first one, adjacent to the card file station, allows for a semi-trailer to be loaded from the side, which is efficient and saves time since the entire length of the semi-trailer can be exposed. The other gate is a traditional loading bay where lorries and semi-trailers are backed up against a platform for loading from the back door. This gate can also accommodate the loading of ISO-containers.

Adjacent to the card file station, two rows of shelves, 10 stories high, are served by an automatic high-stacker forklift. This forklift is operated by the card file operator. Pallets on the conveyor can be picked up by the high-stacker and put into these shelves (or the high-stacker can fetch pallets to place on the conveyor belt).
The rest of the warehouse consists of shelves 5 floors in height. There are three types of forklifts in use:

- high-lifting forklifts that can lift pallets 5 floors
- powered manual forklifts that can reach ground levels only
- manual forklifts used for smaller movements on the ground

The shelves are classified according to what types of goods they may or may not contain. Some of the shelves are for instance reserved for flammable goods, while some are reserved for other dangerous goods classes.

The first floor of all the five-story shelves (the ground floor) are reserved for consolidated less-than-pallet orders. In these locations, all pallets are placed according to their product number, and the shrink-wrap is removed, thus enabling forklift drivers to pick individual products when filling smaller orders. The locations for each product are never changed, as the picking forklift drivers soon learn where to find the various products.

5.6.5 Heterogeneous goods

All the goods handled in this case are loaded on pallets or can otherwise be handled with forklifts. Some of the products are flammable or corrosive and must be stored separately, but since this is a paint factory, several of the products are classified as dangerous and are therefore not considered heterogeneous. A picking order, for instance, may become heterogeneous if the card file system does not contain the right information or if the pallet is placed in the wrong slot. If that is the case, the pallet must be located manually. There are several ways for a pallet full of products to effectively disappear inside this system because of the way the information system is designed.

5.6.6 Case chronology

Since this case does not depict a single transportation chain, there is also no single chain of events to describe it. The case chronology will for this case instead consist of smaller, more isolated, chronologies of each of the studied processes within the warehouse operation.

Receiving pallet from factory

In this process, a pallet arrives from the adjacent factory and is placed in the warehouse.

1. The pallet is loaded by factory personnel.
2. The pallet is shrink-wrapped in plastic film.
3. The factory personnel prints an internal waybill and places two copies of this on the pallet.
4. The factory personnel places the pallet on one of the four wagons of the AGV-train.
5. When the train is fully loaded, it departs for the trip to the conveyor belt (about 100 metres).
6. The wagons are automatically unloaded onto the conveyor belt (the pallets “slide” off the wagons onto the conveyor belt).
7. The conveyor belt transports the pallets to the card file station.
8. When a pallet reaches the card file station, the card file operator takes one of the internal waybill copies.

9. Depending on the product code on the waybill, the card file operator assigns the pallet to a specific slot in the warehouse by selecting a plastic card of the right colour from a collection of free cards. There is one card for each slot in the warehouse and the colour of the cards represents the type of goods that the slot is allowed to store (flammable etc.). The card is of plastic and measures 10*20 cm. Printed on the card is a coordinate denoting shelf number, column number, and floor.

10. The copy of the internal waybill that the card file operator took from the pallet is folded around the selected placement card and filed in product code order in the card file. The coordinate printed on the card is written with a black marker on the plastic shrink-wrap film that surrounds the pallet.

11. If the pallet was assigned to the high-stacker shelves, the card file operator lets the high-stacker forklift collect the pallet from the conveyor belt for placement in the high-stacker grid. The placement code is entered in a dedicated computer terminal that controls the high-stacker forklift.

12. If the pallet was assigned to other shelves (with 5 floors instead of 10), the conveyor belt is moved forward so that the pallet is placed at the very end of the belt.

13. A forklift operator collects the pallet from the conveyor belt and transports it to the assigned coordinates.

14. In some cases, the pallet is assigned to floor 1 (the ground floor) of a shelf. These pallets are opened up, i.e. the shrink-wrap plastic film is removed. Pallets like these are used when consolidated orders are picked.

Fetching a full pallet from the warehouse

In this process, a full (still shrink-wrapped) pallet is retrieved from the warehouse.

1. The card file operator receives an order to fetch a pallet of a certain product.

2. In the card file, each pallet slot is represented by a plastic card. They are sorted according to product number. The card file operator finds the correct product number in the card file.

3. With each card is stored an internal waybill that represents the contents of the pallet. Among other things, this waybill contains production date. The card file operator chooses the oldest pallet containing the wanted product.

4. The coordinates for the correct slot is written on the internal waybill by the card file operator. The card file operator also writes where to put the fetched pallet.

5. The waybill with the entered coordinates is placed in a box outside the card file station where a forklift operator can fetch it.

6. A forklift operator fetches the pallet and puts it in the designated place.
Picking a consolidated order

Several orders from customers are of smaller quantities than full pallets. In this process, a forklift driver collects products from several slots and consolidates them into a single consignment.

1. A customer places an order, either via phone, EDI, e-mail or fax.
2. The order is entered into a computer system.
3. A picking list is printed containing each item in the order. The list is not sorted in any special way.
4. The picking list is placed in a document basket. In the terminal, there are several open spaces, called plazas that are used for intermediary storage of consignments that are to be shipped. Each plaza has its own basket.
5. A forklift driver who is free, takes the topmost picking list.
6. The forklift driver picks up an empty pallet.
7. The forklift driver picks all the items on the picking list.
8. If the list contains loose products, like spray cans or paint buckets, the forklift driver places everything into a box. When a pallet is loaded, it is shrink-wrapped and placed on the designated plaza.
9. The forklift driver certifies in the computer system that the order has been picked (noting any backlogged items).
10. The forklift driver prints freight documents and places these on the pallet.
11. The picking list is archived for future reference.

5.6.7 Case conclusions

This case clearly illustrates the uses of high quality information and the drawbacks of an inadequate information system. The card file is an extremely work-intensive, error-prone, and inefficient solution to the information handling problem in the warehouse. To perform the necessary interactions, the operator in the card file must possess not only a high degree of specialised knowledge but also thoroughness, not often called for in operations of this type. There are several possible sources of errors during the card file operation:

- The wrong waybill may be associated with the pallet
- The wrong coordinates may be written on the pallet
- The card may be placed in the wrong place in the card file, i.e. among cards belonging to other products
- The card file operator may write the wrong coordinates on the pallet
- The forklift operator may place the pallet in the wrong slot (this includes the high-stacker forklift)

The consequence of each of these events is that the pallet effectively disappears. If the card file cannot be used to find it, it must be looked for manually, either by searching the entire warehouse, or by sifting through the entire card file to find a misplaced card.

According to the manager, almost one whole full-time employee is needed to correct errors that are made due to the card file system.
5.7 Case 7 – Transportation and handling of computers in a road transport terminal

The same terminal as in Case 1 acts as a national hub for a large computer manufacturer. This manufacturer has a number of requirements for the transport and handling of their goods. Since the goods are extremely sensitive to theft, all handling takes place inside a caged area of the terminal. Semi-trailers arrive in pairs from the manufacturer, loaded without the use of pallets to maximise utilisation. Terminal operators unload the semi-trailer and put the packages in pre-defined slots on the terminal floor. Each slot number corresponds to a number given on the consignor label attached to each package. As soon as the goods are unloaded from the semi-trailer, an operator scans a bar code on each package. This information is then checked against a pre-notification sent by the computer manufacturer in advance. When all the goods have been unloaded, each slot on the terminal floor that contains goods should now have a complete consignment. Each consignment is loaded onto a pallet, overpacked (i.e. wrapped in plastic film) and sent to its destination terminal using the same network as described in Case 1.

5.7.1 Case objective

This study was conducted in order to examine the extra interaction required between the forwarder and a consignment from a certain customer – in this case a large computer manufacturer. The normal operation for this forwarder (as described in Case 1) does not include many of the restrictions imposed upon the consignment belonging to the computer manufacturer. This study is a comparison with Case 1.

5.7.2 Case methodology

The data for this study has several sources. Some of it comes from first-hand observations and interviews. Some of the data comes from student projects carried out in a course on Logistics Development at Chalmers University of Technology (Ax et al., 1999; Carlsson et al., 1999). All the data taken from the student reports has been double-checked with the forwarding company.

5.7.3 Actors and roles

This case, as in case 6, describes an internal operation. The list of actors is therefore limited to the various roles found within the scope of the case study:
Table 21 Actors and roles in case 7

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarder</td>
<td>Delivers the trailer to the terminal for unloading</td>
</tr>
<tr>
<td>Terminal operator</td>
<td>Unloads and sorts the goods, creating consignments</td>
</tr>
</tbody>
</table>

5.7.4 System description

Within the terminal described in Case 1, there is a caged area. The area encompasses two of the incoming gates as well as a large space inside these gates. Each gate has been equipped with a non-motorised roller conveyor (storage area). Next to the gates are temporary storage areas. Further into the terminal is a grid painted on the floor (250 squares all together). Each square contains four numbers:

```
000 001 002  ETC.
250 251 252  ETC.
500 501 502  ETC.
750 751 752  ETC.
```

*Figure 48 Layout of the sorting grid. There are 250 squares, each containing four numbers as shown in the figure.*

Each square in the sorting grid is large enough to contain several consignments.

There are several points of difference between this case and Case 1:

- The goods are not loaded on pallets.
- Packages belonging to the same outgoing consignment may arrive in different semi-trailers, on different days even.
- The arrival rate varies between 2 and 30 semi-trailers per day.
- The customer has specific demands on traceability of each package, a requirement that cannot be met using the forwarder’s standard information architecture for palleted goods.
- The forwarder performs a merge-in-transit operation on behalf of the computer manufacturer. In a separate warehouse – 5 minutes by car – the forwarder maintains a warehouse operation. This warehouse contains computer screens (a few models). If a customer in the forwarder’s region has ordered a computer with one of these screens, the forwarder is responsible for the delivery of the right screen together with the rest of the consignment. A consignment holds an average of between 3 and 4 separate items.
The sorting grid is used to collect complete consignments. As semi-trailers arrive from the computer manufacturer, they are unloaded manually onto the conveyor. Each box is scanned and checked against a manifest. All boxes belonging to a single consignment share the same order number on their address labels. Each box is placed in the grid square with the number matching the last three digits of the order number. Since the order numbers can be assumed to be evenly distributed, the grid can handle a large number of consignments.

Once a consignment is complete, it is shipped to the consignee using the forwarder’s own network (with some exceptions).

5.7.5 Heterogeneous goods

Every single box in this case is heterogeneous. Not in physical measurements, but in their information content and in their handling interface, i.e. they have to be unloaded manually. When a consignment is assembled and transferred into the “normal” system of terminals, this is still heterogeneous because of special handling instructions and track-and-trace demands.
5.7.6 Case chronology

1. A manifest arrives by e-mail or fax to the forwarder. It contains information on the contents of a forthcoming semi-trailer.

2. If a merge-in-transit is required, an order is placed at the warehouse for delivery to the terminal.

3. The semi-trailer arrives. The semi-trailer is loaded floor to ceiling with cardboard boxes. No pallets are used.

4. The semi-trailer is unloaded manually.

5. Each box is placed on the roller conveyor and scanned.

6. If the box was not included in the manifest or for some other reason needs attention, the computer system warns the operator scanning the boxes. In these cases, the box is removed from the conveyor and handled separately.

7. A terminal operator takes each box from the roller conveyor and places it in the grid. The last three digits on the box’s order number give the number of the square to place the box in.

8. Once every box is checked, the semi-trailer is allowed to leave.

9. After the unloading is finished, any boxes not placed in the grid are handled.

10. The grid should now contain all the boxes, and parts belonging to the same consignment should be found inside the same square.

11. A terminal operator controls the consignments in the grid. If a consignment is not complete, he can do the following:
   a. If an item that should have been merged is missing, he can remedy that himself.
   b. If another item is missing, he can either let the consignment wait in the grid for the next semi-trailer or put the incomplete collection of items in a certain observation slot. If it is put in such a slot, the computer system will be flagged so that when a missing item is found in the arrival scan, it is taken directly to the rest of the consignment.

12. If a consignment is complete, it is placed on a pallet together with other consignments going to the same destination terminal (that will have to sort the items again). Consignments that are to be distributed from this terminal are placed on individual pallets directly.

13. Once a pallet is full, it is taken to the destination gate (a gate corresponding with the destination terminal).

14. The pallet is loaded onto a semi-trailer in the same fashion as described in Case 1.
15. The handling of the pallet is exactly like described in Case 1 except when delivering to consignee:
   a. The delivery must take place during the day.
   b. If the delivery is not during the day, the driver must have company.
   c. The consignment must be scanned on delivery. At the time of this study, the delivering lorries did not have this equipment. The forwarder had the technology, although it was only in use within their parcel service (items < 50 kg). The lorries making these deliveries had therefore to be outfitted with scanning devices.

5.7.7 Case conclusions

There are several striking differences between Case 7 and Case 1. The operation described in Case 1 was deliberately almost void of interaction between consignment and TCS. In this case, however, there is far more interaction. The goods arrive unsorted, leading to a comprehensive and labour-intense sorting operation. The goods are theft-prone, leading to high demands for security and traceability. The individual consignment needs to be considered and processed on numerous occasions, leading to high frequency of the interaction.

5.8 Case 8 - Components for automotive assembly

In this case, a car manufacturer orders assembly parts from a sub-contractor. The flow is regular and all of the involved actors are familiar with the layout of the system. This case focuses on the forwarding company, and on the administration surrounding the studied flow.

5.8.1 Case objective

This case study was originally done as a means of testing object-oriented modelling. The objective was to study how a highly frequent and well-trimmed flow is operated on a high TCS level. The study is here used to investigate how much interaction a well-designed flow has between the top-level TCS and the consignments. Top-level means that the TCS is not physically in contact with the consignments and that the interactions are on the information level.

5.8.2 Case methodology

The interviews in this case have been conducted somewhat differently than with the other cases. The goods flow is not studied directly, but rather indirectly through information flows. The interviewed actors have all been administrative personnel with forwarding responsibilities. The event-driven selection of interview objects have been applied also in this case, although it is the information flows that have been followed.
5.8.3 Actors and roles

The following actors were identified:

Table 22 Actors and roles in case 7

<table>
<thead>
<tr>
<th>Actor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consignor</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; tier supplier to the automotive industry. The company is situated in France.</td>
</tr>
<tr>
<td>Consignee</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; tier supplier to a car manufacturer. The consignee assembles subsystems for the main car assembly plant that lies a few kilometres away.</td>
</tr>
<tr>
<td>Forwarding company, sending side</td>
<td>Situated in Belgium, part of the same company as the forwarding company on the receiving side. Operates a terminal for sending and collecting consolidated goods.</td>
</tr>
<tr>
<td>Forwarding company, receiving side</td>
<td>Situated close to the consignee. Operates a terminal for receiving and dispersing consolidated goods.</td>
</tr>
<tr>
<td>Swedish road haulier</td>
<td>Transports the consignment from the ferry to the consignor, sometimes via a dispersing terminal.</td>
</tr>
<tr>
<td>Ferry operator</td>
<td>Runs a ferry line between Sweden and Belgium.</td>
</tr>
<tr>
<td>Road haulier on the sending side</td>
<td>Transports consignment (semi-trailer) from the consignor to the ferry, sometimes via a collecting terminal.</td>
</tr>
</tbody>
</table>

5.8.4 System description

The goods flow in this case is quite uncomplicated. Every 2.5 days a consignment of components is loaded in France, transported to the consignee by the exact same route every time and handled by the same people. The information exchange is standardised and the consignor as well as the consignee and the forwarding companies have computerised equipment at their disposal to handle any insecurity. According to the interview objects, this case adequately represents a typical automotive industry supplier relationship.
Statistical data including all consignments between the consignor and the consignee during 26 weeks of the studied year has been obtained. A total of 109 consignments took place during the period, which makes an average of 4.19 consignments per week. The system is designed for 2.5 consignments per week (alternating 2/week and 3/week). As can be seen in Figure 50 above, there are several consignments that are less-than-full semi-trailer loads (25 tons). Consignments below 18 tons are normally as part load or general cargo routed via the collecting and dispersing terminals for consolidation.

### 5.8.5 Heterogeneous goods

Every product that is transported in this case is loaded onto standardised unit loads. The flow is repetitive, and the involved actors are familiar with the operations. The frequent transports with low load factor are a disturbance that may cause these consignments to be considered heterogeneous along the trajectory.

### 5.8.6 Case chronology

1. Order products. The consignee has access to the production plan of its customer. By using this plan, together with the inventory database, the consignor can calculate the demand of components needed for production. When this calculation is made, an order is sent to the consignor via EDI. The message contains the production plan where the wanted quantities are flagged for delivery on a specific date.

2. Shipping of products. The consignor receives an order from the consignee. The order is requested from the ERP system. Based on the contents of the order, a Customised Transportation Document (CTD)\(^{16}\) is created. The car manufacturer demands that all suppliers use this document when sending goods to them. One CTD can contain

---

\(^{16}\) The CTD is created by the car manufacturer and has been renamed.
several consignments from a single consignor. The CTD is created manually (from a template in MS Word) and faxed to the transport company. The transporting company manually adds the information in the CTD into the sending forwarding company’s transport database. This database contains information on all the transports in their system and can be accessed by the receiving forwarding company. This serves as a notification to the receiving forwarding company that goods will arrive. The consignor prepares the consignment for shipment (attaching freight documents etc). The transporter arrives with a semi-trailer. The consignor loads the consignment on the semi-trailer.

3. Consignment received at collecting terminal. Depending on the size of the consignment, the sending forwarding company either orders the transporter to drive directly to the ferry, or to go via the goods terminal operated by the forwarding company (here called the collecting terminal). If there is more room in the semi-trailer, other consignments can be consolidated with the original one at the terminal. If this is the case, the record in the transport database representing the consignment is altered to display the other contents of the semi-trailer as well. After the consolidation is finished, the transporter drives to the ferry.

4. Consignment sent with ferry. The ferry is loaded by the ferry operator and when the loading is finished, the sending forwarding company sends an e-mail containing the contents of the ferry to the receiving forwarder company. The ferry sails at 3 am and takes 42 hours to reach the destination port (arrives at 9 pm).

5. Consignment received at the dispersing terminal. The sending forwarding company sends a list of the contents of the ferry to the receiving forwarding company. The receiving forwarding company checks with the transport database to see if the semi-trailer contains more than one consignment. The ferry arrives and is unloaded by the ferry operator. If the semi-trailer does not contain consignments to more than one company, the transport operator on the receiving side is ordered to drive directly to the consignee. If, on the other hand, the semi-trailer contains consolidated consignments, the transport operator is ordered to drive to the dispersing terminal.

6. Goods de-consolidated in the dispersing terminal. At the dispersing terminal, the consignments are de-consolidated by the forwarder. When this is finished, the transporter is ordered to drive to the consignee.

7. The consignment is transported to the consignee.

8. The consignment is received. The consignee gets notified that the consignment has arrived. The consignee unloads the consignment and updates the inventory database. When the unloading is finished, the transport operator departs.

5.8.7 Case conclusions

This system is, according to the interview objects, typical for the automotive industry. The studied system has a few weak points, though:

- Some of the critical information operations are performed manually.
- The system is bureaucratic in nature, giving individual actors little or no insight into the activities of others.
- The performance rating is low regarding load factors and transport cost for the consignee.
As can be seen in the figure above, most consignments are below the 18 tonne limit, thereby regarded as LTL transports. The load factor is calculated as the weight of each consignment divided by the max load of the semi-trailers (25 tonnes). 25% of the transports qualify as FTL, carrying a fixed price. 75% are classified as LTL, carrying a price per transported weight. Notable is also that 40% of the total number of consignments and more than half of the LTL transports have a less than 50% load factor.

### 5.9 Summary of the case studies

Eight cases have been studied with the overall motive of searching for evidence supporting the claim that heterogeneous goods increase complexity and thereby, cost. The cases were chosen partly for their fitting into the structure presented in Figure 40 on page 90. Partly they were selected by circumstance, as they – for practical reasons – fitted into already existing projects. Regardless of how they were chosen, they were studied with the same basic motive and using the same basic data collection method – interviews together with observations.
<table>
<thead>
<tr>
<th>Case</th>
<th>Goods type</th>
<th>Approx. shipment frequency</th>
<th>Type of load unit used</th>
<th>Transport modes involved</th>
<th>Terminals involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Packaged goods on pallets</td>
<td>&gt;1700/day</td>
<td>Pallet, Semi-trailer</td>
<td>Road</td>
<td>Two domestic road network terminals</td>
</tr>
<tr>
<td>2</td>
<td>Packaged goods Each transport unique</td>
<td></td>
<td>Container</td>
<td>Road, Sea</td>
<td>Domestic road terminal, port</td>
</tr>
<tr>
<td>3</td>
<td>Liquid</td>
<td>2/week</td>
<td>Tank container</td>
<td>Road, Sea, Rail</td>
<td>Ferry port, intermodal terminal</td>
</tr>
<tr>
<td>4</td>
<td>Large rolls</td>
<td>2/week</td>
<td>Semi-trailer</td>
<td>Road, Sea, Rail</td>
<td>Ferry port</td>
</tr>
<tr>
<td>5</td>
<td>Large rolls</td>
<td>2/week</td>
<td>Semi-trailer</td>
<td>Road, Rail, Sea</td>
<td>Intermodal terminal, ferry port</td>
</tr>
<tr>
<td>6</td>
<td>Packaged goods on pallets</td>
<td>&gt;200/day</td>
<td>Pallet</td>
<td>Intra terminal, road, sea</td>
<td>Production warehouse acting as national hub, foreign terminal</td>
</tr>
<tr>
<td>7</td>
<td>Computers in boxes</td>
<td>&gt;100/day</td>
<td>None</td>
<td>Road</td>
<td>Domestic road network terminal</td>
</tr>
<tr>
<td>8</td>
<td>Components for automotive assembly</td>
<td>3/week</td>
<td>Pallet, Semi-trailer</td>
<td>Road, Sea</td>
<td>Two road terminals, Ferry port</td>
</tr>
</tbody>
</table>

The goods types range from homogeneous to heterogeneous. Case 1 describes already consolidated goods in a road network terminal. The only handling equipment needed is a forklift truck. Case 2 describes containers that are packed by the forwarder. Thus, the contents of each container differ from transport to transport. Also, each container consists of several consolidated consignments and is therefore subject to attention requirements from several consignors/consignees. Case 3 is the only case where a bulk transport is described. In this particular system, 80% of all transports contain dangerous goods, and are therefore subject to several regulations. The goods type in Cases 4 and 5 is given as “large rolls”. Case 4 describes rolls of wall-to-wall carpets, and Case 5 rolls of paper. Common for them is the fact that the rolls cannot be handled in a standardised fashion as with pallets. Case 6 describes several goods types, ranging from paints to adhesives and solvents, several of them regarded as dangerous goods. All of the goods are consolidated on pallets, though, enabling standardised handling equipment. Case 7 takes place in the same terminal network as Case 1. The forwarder has a special client (computer distribution company) with demands that their goods be treated separately. The difference between the goods in this case and the goods from Case
1 mainly lies in the fact that the goods in Case 7 have greater demands on the information system. Case 8 depicts a typical transportation system for the automotive industry, with frequent and standardised deliveries of components.
6. ANALYSIS

This chapter contains analyses of the empirical data from the cases coupled with the theory within the frame of reference. The analysis in this chapter is focused on the first two research questions:

Q1 How does the interface of the consignment affect the complexity drivers during a transportation process?

Q2 How does the interface of the Transport Control System affect the complexity drivers during a transportation process?

The analysis process is presented in chapter 2.3.2. The first step is to construct OO-based models of the cases. In the process of doing this, a class library has emerged which is presented in the next section. The second step is to study the OO-based models of the cases and perform cross-case comparisons focused on the Consignment objects and the TCS objects respectively. The research questions Q1 and Q2 will be addressed in this step. Step three involves a calculation example where data from one of the cases is used to construct an OO-based model that can be manipulated in a simulation program, calculating results based upon changes in the model. Q3 is addressed in the next chapter using the analysis from this chapter as an input.

This chapter concludes with a section examining the research as such in terms of validity and reliability.

6.1 Step one: Building object models

The first step in the analysis is to process the empirical data through the OO-framework. For each of the eight cases, diagrams of four types have been constructed (class diagrams, use case diagrams, sequence diagrams, and statecharts). All the diagrams can be found in Appendix 5.

6.1.1 The modelling process

The modelling process is iterative. Once a diagram is created, it can be revised and extended as the model grows. This ongoing refinement can be seen as delving into lower and lower levels of abstraction where each new level includes new classes, use cases, sequences, and trajectories. In the modelling of the cases in this thesis it has been important to stop this refinement in time so as to not go beyond the scope of the empirical data gathered. For some cases, the abstraction level is higher (cases 2, 3, 4, 5 and 8) and for some it is lower (cases 1, 6, and 7).

The details of the modelling process have been similar between the cases. The principal relationship between the real-world case data and the various diagrams are depicted in the class diagram below:
First, general information about each case is assembled. This acts as a backdrop for the construction of the diagrams.

Next, the actors and their roles are identified. Together with a list of entities, the events can be listed in chronological order. Each event may encompass one or more actors and one or more entities.

The OO-based model begins with creating class diagrams from the lists of actors and entities. The creation takes support from the general description in describing any attributes and operations that are identified in the class structure.

Next, the use cases are identified. By studying the list of events, together with the list of actors, sequence diagrams can be constructed, and following these, statecharts.

Once the first version of a diagram is finished, the process is reiterated with the other diagrams, extending and redesigning these as needed to encompass all the data that is present in the case description.

The diagrams for the case studies are found in their entirety in Appendix 5.

### 6.1.2 Class library

The static structures of the cases (or parts of the cases) are presented in class diagrams. Since all of the studied cases are in some way aspects of the same kind of system, several of the classes that have been identified have been found in several, if not all, of the cases. One of the strengths in object-oriented modelling is the reusability where a class, once defined, may be
reused in other models. Therefore, the process of constructing class diagrams of the cases has yielded a library of generic classes and relationships that are present in all of the cases.

Below is the generic class Load unit:

![Diagram of the generic class Load unit](image)

*Figure 53 The generic class Load unit.*

When using the class Tank container when describing a case, for instance, the class diagram above provides the information that the attributes consist of Size, Weight, Position, BIC code, and Volume. The sub-classes inherit the interface structure of its parent. When building OO-models, such generic classes save effort and help the modeller keep consistency between cases.
Below is the generic relationship between TCS, Consignment and Load unit:

![Diagram of TCS, Consignment, and Load unit]

In the diagram above, it is clear that a TCS may have access to several (denoted by *) Consignments. Each Consignment may be associated with a number of Load units (or none). Also, each consignment contains multiple pieces of information, which can be either physical, or electronic. The Consignment must be created by an Actor.

The Actor is also a generic class:

![Diagram of Actor]

As can be seen in the diagram above, the Actor can be either a Person or a Company. Each Actor can play several roles such as Consignor, Consignee, TCS etc. In the previous diagram (Figure 54), the TCS-role is presented as a stand-alone class. This is done to clarify the relationship with an Actor playing the role of TCS and a Consignment.
The TCS, as is shown in the model in Chapter 4.1.1, controls resources. The Resource class is defined below:

![Diagram of Resource class hierarchy]

Figure 56 Generic class diagram of a Resource.

The Resource class has, as with all the generic classes in this thesis, been constructed using data from the cases. It is therefore possible, even likely, that the hierarchy of resource types needs extension. This is not a problem however, since a new type of Handling equipment, for instance, does not disturb previous models. It merely extends the class library by adding types while keeping it “backwards compatible”\(^\text{17}\) with older models based on it.

---

\(^{17}\) This is a term often used when describing computer programs that can handle interfaces created by future versions such as when the previous versions of Microsoft Word are able to open and edit documents created in the latest version.
The last class in the library is the Facility:

![Generic class diagram of a Facility.](image)

A facility can, based on the cases in this thesis, be either a Loading dock, Warehouse, Terminal etc. It has Operators and Resources and may also have an Address.

This class library has been used when constructing the models in this thesis. When needed, the generic classes have been extended, see for instance Diagram 23 in Appendix 5, to encompass attributes or subtypes relevant to the particular case.

### 6.2 Step two: Cross-case comparisons

In this analysis step, the cases are studied, compared and analysed. The OO-diagrams have been used as a reference and can be found in their entirety in Appendix 5. There are two main cross-sections of the case data.

First, the cases are compared regarding the control of the transportation process and how it is affected by the complexity drivers. Second, the TCSs identified in the cases are compared regarding their various strategies in minimising the effects of the three complexity drivers.

#### 6.2.1 Controlling the transportation process

As was stated in Chapter 4.4, the control of the transportation process takes place in three different time-spans – long-term, medium-term, and short-term.

In the long-term control scope, the primary task is to design the system in such a way that the descriptive complexity is reduced.

In the medium-term control scope, the primary task is to reduce the computational complexity with either design, or by manipulating the system in advance so that the number of trajectories decreases before the actual transportation process begins.

The short-term control aims at reducing the uncertainty-based complexity, mainly by manipulating the interface of the consignment in direction of the goal state. In some cases, the long- and medium-term control has reduced the complexity to such a degree that the short-term control consists of monitoring the trajectory to ensure that it follows a predefined path.
In Table 24 below, the three control scopes are listed for each of the studied cases:

*Table 24 How the control scopes are manifested in the cases, and what primary task was observed for each scope in each case.*

<table>
<thead>
<tr>
<th>Case</th>
<th>Long-term control</th>
<th>Medium-term control</th>
<th>Short-term control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Task: Design</strong></td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Monitor</strong></td>
</tr>
<tr>
<td></td>
<td>The design of the system facilitates the medium- to short-term control</td>
<td>No explicit medium-term control was observed.</td>
<td>The number of trajectories is relatively small which in turn lessens the number of short-term decisions.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Design</strong></td>
<td><strong>Task: Design/operate</strong></td>
</tr>
<tr>
<td></td>
<td>Many decisions were delegated to the medium- and short-term control</td>
<td>The design of the system depends on current status of for instance shipping lines</td>
<td>Most of the decisions were made in this time span.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Task: Design</strong></td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Monitor</strong></td>
</tr>
<tr>
<td></td>
<td>The design of the system facilitates the medium- to short-term control</td>
<td>No explicit medium-term control was observed</td>
<td>The studied transport is of a dedicated container, and few decisions have to be made during the transportation process.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Design/operate</strong></td>
<td><strong>Task: Operate</strong></td>
</tr>
<tr>
<td></td>
<td>Many decisions were delegated to the medium- and short-term control</td>
<td>Resource allocation was made in anticipation of future transports (empty semi-trailers were positioned close to the consignor).</td>
<td>Some decisions were made during, or close to, the transportation process.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Design/operate</strong></td>
<td><strong>Task: Operate</strong></td>
</tr>
<tr>
<td></td>
<td>Many decisions were delegated to the medium- and short-term control</td>
<td>Resource allocation was made in anticipation of future transports (empty semi-trailers were positioned close to the consignor).</td>
<td>In the studied case, the trailer missed the ferry, causing a large number of decisions during the transportation process.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Task: Design</strong></td>
<td><strong>Task: None</strong></td>
<td><strong>Task: Operate</strong></td>
</tr>
<tr>
<td></td>
<td>The design of the system facilitates the medium- to short-term control</td>
<td>No explicit medium-term control was observed</td>
<td>In the studied warehouse, several decisions are made, especially by the card file operator</td>
</tr>
<tr>
<td>Case</td>
<td>Long-term control</td>
<td>Medium-term control</td>
<td>Short-term control</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>7</td>
<td>Task: Design</td>
<td>Task: None</td>
<td>Task: Operate</td>
</tr>
<tr>
<td></td>
<td>The design of the system facilitates the medium to short-term control</td>
<td>No explicit medium-term control was observed</td>
<td>Each consignment must be handled and evaluated several times</td>
</tr>
</tbody>
</table>

| 8    | Task: Design      | Task: None          | Task: Monitor      |
|      | The design of the system facilitates the medium to short-term control | No explicit medium-term control was observed | The number of trajectories is relatively small which in turn lessens the number of short-term decisions. |

In the following sections, examples from the cases are presented, partitioned for the three complexity types.

**Reducing descriptive complexity through design**

In the long-term control scope, the descriptive complexity can be reduced through systems design. The descriptive complexity denotes the number of possible states the system can assume. The larger the state space, the more choices must be made when planning a transportation process. For homogeneous goods, the TCS can generalise and disregard certain attributes and their value ranges as being irrelevant to the transformation. This simplification strategy is described in Chapter 3.3.1 as the exclusion of variables. Due to the nature of heterogeneous goods, some of these excluded variables need to be included in order for the TCS to reach a canonical representation of the consignment. In turn, other variables may instead be excluded. It is in fact possible that a consignment with heterogeneous goods may carry a completely different set of attributes and operations (compared to homogeneous goods) needed to facilitate its trajectory in a system.

In case 1, almost every consignment is considered homogeneous and can be treated the same way, regardless of how many gates or consignments are present in the system. When compared to case 7, where two of the gates have been reconstructed to receive heterogeneous goods (computers) it is evident that even though the physical characteristics of the products transported in case 1 are similar to the products in case 7, they are fundamentally different. The consignments in case 7 arrive at the terminal with completely different state spaces than the homogeneous goods normally present in the terminal. In order to handle this variety, the terminal is reconstructed and the handling of the heterogeneous goods is encapsulated into a subsystem that has one goal function: transform the heterogeneous goods enough so that they can be integrated within the homogeneous goods flows. In case 1, the terminal has a coordination desk whose sole task is to designate the consignments to a departure gate. By encapsulating this function to a separate object, the system variety is lowered compared to if all the terminal handlers were to have the freedom to choose departure gates. See Diagram 11 and Diagram 14 in Appendix 5.

Case 2 displayed a big difference in state space when a consignment contained dangerous goods. Information that is specific to the transportation system, i.e. the Dangerous Goods Declaration, had to be produced by the consignor. According to the case study, consignors sometimes made errors in this regard, thereby causing the whole shipment to be delayed.
All of the above depicted documentations are required at some point during the trajectory of the consignment in case 2. The only exception is the PSA-documentation, which is needed if the container is to arrive in Singapore. This documentation is the responsibility of the consignor, and without it, the whole container may be denied service in Singapore. The forwarder’s strategy to avoid this was to take responsibility for the production of these documents from the consignor. Either the forwarder produced the documents himself, or told the consignor to do it.

In case 3, the variety is kept down with efficient encapsulation and application of constraints to ensure that the system operates smoothly. One example can be found when studying the class diagram for the Consignment together with its Sequence diagram. By treating every consignment as if it were dangerous, the system’s variety is reduced. The only difference is found when a container does not carry dangerous goods. Then, certain documents will not be present.

In the cases 4 and 5, the same ferry line is used for transporting the consignments. In one of the cases, case 4, the semi-trailer is loaded directly onto the ferry. In case 5, the semi-trailer is first loaded onto a rail car, then onto the ferry. The semi-trailer from case 4 is loaded onto a railcar after the ferry transport. When studying the ferry operator as a TCS, some differences can be found when comparing the various consignments from cases 4 and 5. Compare Diagram 35 and Diagram 36 (page 228-229) to Diagram 46 and Diagram 47 (page 235-236) to see the difference in the role of the Ferry line. The ferry line clearly has different roles between the cases 4 and 5. In case 4, the ferry line acts as TCS with a responsibility for a semi-trailer during loading, transport, and unloading. In case 5, the ferry line can be seen as just an extension of the rail tracks with no responsibility other than to deliver the ferry (i.e. the rail tracks) to the next node, in this case a port in Germany.

In case 6 only one TCS has been studied. Within the studied warehouse, several different Consignment objects are handled. Consignments arrive at the warehouse in two different ways, either by internal AGV transport, or by external delivery. Once the products have arrived, they are treated in much the same manner and are put into storage for later retrieval. The most sensitive part of the studied warehouse is the card file system. The variety in the card file system is fairly constant as the warehouse has a fixed number of slots, with one card for each slot. The cards are classified into several categories to facilitate easy access to information. By doing this, the variety for each subsystem of cards becomes manageable.

Case 7 takes place in the same terminal as the Collecting terminal from case 1. The variety differs greatly, though. In case 1, the arrival and unloading of consignments was a fairly
straightforward procedure where the variety was kept down due to the fact that almost every consignment was handled in the exact same fashion. This kept the interactions with the individual consignment to a minimum, and the few decisions that had to be made were encapsulated into dedicated subsystems. In case 7, every part of every consignment has to be regulated. The purpose of the arrival process is to reduce the variety by collecting parts of consignments together to be able to encapsulate them as a single object that can be transferred to the standardised process described in case 1. A large amount of manual labour is used to sort the products into consignments.

Case 8, as a part of the automotive industry, is very bureaucratic in its organisation. The interfaces between the actors are well-defined which in turn makes the encapsulation more complete. Because of this, the various TCSs can only get a full view of the state space they themselves control. This is also observed in case 1, between the collecting and dispersing terminals. A strategy like this delegates to every subsystem to handle its own variety with no assistance from the outside. With hierarchically organised TCSs observed in the cases, there is always the option of calling the “parent” TCS when variety becomes too great to handle.

The descriptive complexity that has been observed in the studied cases is primarily handled by encapsulation. The overall strategy that has been observed is to design hierarchies of nested subsystems. These subsystems, or objects, each contain a certain, manageable, variety that can be assigned to another TCS to regulate.

Reducing computational complexity by planning

The computational complexity can be reduced during the medium-term control scope when planning a transportation process. The computational complexity affects the knowledge needed to make the “right” decisions. Even though variety indicates the number of possible states, this is only a theoretical value. To reach the goal state, the system has to pass through a number of transitions, a trajectory. In transportation systems, there are often several valid trajectories to choose from. A valid trajectory is a succession of states that is practically feasible. To recognise a valid trajectory from an invalid one needs knowledge not only about the consignment, but also about the total state space. The TCS must have the ability to choose the “best” trajectory for each consignment.

In case 1, there are not many alternative trajectories present. The system is highly standardised and any irregularities and deviations are handled outside the scope of the study. In case 7, for instance, the consignments are transformed to comply with the interface structure required for the terminal in case 1. The only alternative trajectory is identified in the treatment of dangerous goods, which are required to be placed separately in the terminal and which are loaded separately onto the semi-trailer.
Within the studied scope of case 2, the trajectories of the consignments do not differ much once they enter the terminal of the transporting agent. Even though this is a complicated, information-dense case, the actual flow of goods is quite straightforward. The main differences lie in the choice of shipping line, which in turn is a product of the location of the consignee as well as of the time restrictions for the transport.

In case 3, few alternative trajectories are found. This flow is highly standardised and the only difference that has been observed is the choice of either cleaning the tank when it is unloaded or not, and even this is mostly made by the consignor.

The Main forwarder in case 4 has a few options when assigning a trailer and a lorry to fetch the consignment from the consignee. Otherwise, the choice of trajectory lies with the consignor as transport initiator. The choice of intermodal transport instead of road-based is a matter of cost, as the speed requirements are not critical.

As in case 4, the consignor in case 5 decides the route. The Main forwarder controls all the details, either directly or via a sub-TCS, like the intermodal operator. There are not many alternative trajectories, although the case encompassed a delayed shipment which showed an alternative trajectory for the information flow.

In case 6, the forklift drivers picking consolidated orders continually face trajectory decisions as they receive unsorted picking orders. They apply their knowledge and experience and choose their trajectory based on what they consider to be the most efficient route. Longer lists require more knowledge and better planning skills. Some of the products in the studied warehouse are not placed in numerical order according to product number, but based on other attributes such as size or flammability. These examples of heterogeneous goods require even further knowledge when planning a trajectory.

The trajectories are well-planned in case 7; not on the individual consignment level, since all goods are heterogeneous, but on an aggregated algorithm-level. The precise instructions on how to use the sorting grid, how to handle deviations from the manifest etc. are examples of how all conceivable trajectories are handled in advance by the system designer.

The sending forwarder of case 8 decides whether the shipment is considered LTL or FTL. This influences the future trajectory since the FTL-shipment goes directly to the consignee while LTL-shipments are unloaded in the dispersing terminal. The FTL trajectory contains less operations and less information than the LTL trajectory. The consignor has access to the

Figure 59 There are only two alternative trajectories in the collecting terminal in case 1.
production plan for the consignee, but in spite of this 75% of the shipments in the study are LTL which result in a higher cost and are more complicated to control for the TCS.

Per definition, the overall trajectory is decided upon by the transport initiator, not the TCS. This means that the TCS has limited choices of alternative trajectories since the transport initiator in most cases not only order a goal state, but also puts limits on what constitutes a valid trajectory. The same is true when a TCS hires another TCS to perform part of a transformation. This reasoning gives the conclusion that the higher the level of the TCS, the more alternative trajectories there will be to choose from. Therefore, it can be argued that a large part of the computational complexity is handled outside the TCS where most alternative trajectories are discarded by the transport initiator.

Reducing uncertainty-based complexity by creating order

In the short-term control scope, the uncertainty-based complexity manifests itself through lack of order (entropy). Even in well-planned systems where the same load unit is transported the same route over and over again, as in case 3, there is embedded uncertainty. The uncertainty-based complexity is affected by the fact that a TCS cannot make every decision in advance. There will always be deviations due to unforeseeable circumstances such as weather, mechanical failures, human errors etc. For heterogeneous goods, this uncertainty is based on the fact that the parameters that define the consignment, i.e. its interface, differ from the ones usually encountered. Again, this is inherent in the definition of heterogeneous goods. These parameters need to be factored in when planning a trajectory, but if the TCS has no advance knowledge of the differences in interface this cannot be done. Therefore, some decisions have to be made “run-time”, i.e. during the transformation process itself.

The lorry drivers in case 1 tend to load dangerous goods at the far back of the semi-trailer to facilitate the process of the police inspecting the cargo. The entropy is lowered because the driver has ordered the cargo in such a way that an inspection process becomes more efficient, thereby enabling him to keep his schedule.

In case 2, containers often contain several consignments. Some of these consignments may have attributes that put them in conflict with the others in the container. An example of this is when the Port of Singapore Authority documentation is not provided by the consignor (he may not know that his consignment will be unloaded there), thereby putting the whole container in hiatus until the right documents are produced. This uncertainty is addressed by the main forwarder who ensures that the consignor produces the right documentation.

In case 3, by using a dedicated container, the entropy is kept low. Every transport is handled and administered in the same way, reducing the task during the short-term control to monitoring.

By not securing the load correctly in case 4, the consignor actually adds to the uncertainty. Frequently (10% of the time according to the interviews), the German stevedores have to correct the load securing. Besides cost, this may cause delays, which in turn may cause problems for the consignee.

When studying case 5, the rail wagon carrying the studied semi-trailer missed the ferry. This resulted in messages being passed involving at least six of the actors. The uncertainty this delay caused clearly required explicit attention.

The entropy in case 6 depends on the state of the card file system. There are several uncertainties embedded into that system, requiring the card file operator to frequently scan several of the cards in order to find a certain piece of information. Also, a significant amount of the resources are dedicated to reduce entropy caused by the poor design of the card file system.
Case 7 has the highest degree of uncertainty of all the studied cases. The TCS that receives the consignments has to, during the transformation, do the following:

- Unload each box manually (no unit loads). Each box must be scanned with a bar code scanner to update its status in the track-and-trace system
- Evaluate information on each box, and depending on this information, make a decision on where to place the box
- Periodically check to see if the sorted boxes have formed complete consignments and if so:
  - Consolidate the boxes on a single pallet if possible
  - Report to the track-and-trace system that the consignment is complete
  - Transfer the consignment to a destination-specific holding area to obtain more space for additional sorting
- Transfer the complete consignments into the main terminal operation, thus integrating with the system studied in case 1.

By allowing the consignments these wide interfaces, the uncertainty – entropy – in the system reaches almost unmanageable levels. In this system, most of the consignments contain heterogeneous goods in the respect that there is no possibility to actually plan in advance on how to facilitate the transformation of a single consignment.

In case 8, the consignor has access to the production plan of the consignee. With this information, the consignor can control the production of products according to the advance knowledge of needs of the consignee. This means that the entropy should be relatively low in this case. This does not explain, however, the low efficiency in the system (as has been stated earlier).

Uncertainty depends on the TCS’s ability to resolve the state space of the system it controls. In some cases, certain states and parts of trajectories are not visible to the TCS, thereby causing uncertainty on cause and effect in the system. If this is the case, there are only two ways for a TCS to remove this uncertainty:

1. Step into the system and observe first hand in order to make informed decisions, thereby resolve the uncertainty
2. Employ another TCS with more knowledge to resolve the uncertainty

Examples of uncertainty have been found in all of the cases. In some, the uncertainty was higher (cases 4, 6, and 7).

6.2.2 Strategies used by TCSs to reduce complexity

It is up to the TCS to reduce the complexity in the transportation system. This can be done within the three control scopes, as was stated in the previous section. In Table 25 below is a summary of how complexity caused by the consignment is reduced by the TCS in each case.
Table 25 How complexity caused by the consignment is reduced by the TCS

<table>
<thead>
<tr>
<th>Case</th>
<th>Reducing descriptive complexity by designing (long-term)</th>
<th>Reducing computational complexity by planning (medium-term)</th>
<th>Reducing uncertainty-based complexity by creating order (short-term)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encapsulation (pallets in semi-trailer)</td>
<td>Skilled subsystem decides</td>
<td>Separate handling of dangerous goods to reduce entropy</td>
</tr>
<tr>
<td>2</td>
<td>Encapsulation (ISO-container)</td>
<td>Main TCS decides</td>
<td>TCS ensures that the consignor produces the right documents</td>
</tr>
<tr>
<td>3</td>
<td>ISO tank container</td>
<td>Consignor decides</td>
<td>Use of a dedicated container</td>
</tr>
<tr>
<td>4</td>
<td>Encapsulation (semi-trailer)</td>
<td>Consignor decides</td>
<td>Poor load securing by the consignor creates uncertainty that requires a sub-TCS to reload the consignment</td>
</tr>
<tr>
<td>5</td>
<td>Encapsulation (semi-trailer)</td>
<td>Consignor decides</td>
<td>A missed ferry departure resulted in intense message passing between several of the actors involved</td>
</tr>
<tr>
<td>6</td>
<td>Hierarchical division of product types in card file</td>
<td>Skilled subsystem decides</td>
<td>The card file has large uncertainty built into its design. This is handled by constantly auditing the contents of the card file.</td>
</tr>
<tr>
<td>7</td>
<td>State space is well-defined by design (sorting grid)</td>
<td>Designed into the system (sorting grid)</td>
<td>High entropy is handled by a multitude of skilled subsystems used to sort parts of consignments.</td>
</tr>
<tr>
<td>8</td>
<td>Encapsulation (pallets in semi-trailer)</td>
<td>Sending forwarder decides</td>
<td>Information about future transport need is shared throughout the chain.</td>
</tr>
</tbody>
</table>

In some of the cases, extensive design reduced the need for medium- and/or short-term control (cases 1, 3, 6, 7, 8). In cases 2, 4, and 5, the planning helped reduce computational complexity. The uncertainty-based complexity was more visible in some of the cases (cases 4, 5, 6, and 7).
6.3 Interfaces
This section examines the interfaces of the consignment objects and of the TCSs respectively. For the consignments, the definition of heterogeneous goods is revisited and each case is examined regarding the control strategy used. For the TCSs, the section looks into the various intra-TCS relationships and how they affect the control of the transportation system.

6.3.1 The interface of the Consignment
In the studied cases, several different Consignment objects have been identified, ranging from small boxes and individual products (cases 6 and 7) to pallets (cases 1, 6 and 8), containers (cases 2 and 3), and FTL semi-trailers (cases 4 and 5).

By identifying the smallest unit of consolidation of the studied consignments, the requirements for handling equipment can be found. In cases 1, 6, and 8, the smallest consolidation unit is the pallet, thereby requiring a forklift for handling. In cases 2, 4, 5, and 7, the smallest consolidation unit consisted of non-standardised interfaces. To overcome this obstacle, the consignments were encapsulated into larger unit loads such as pallets (case 7), ISO-containers (case 2) or semi-trailers (cases 4, and 5). In case 3, the smallest unit of consolidation is the tank container that is used throughout the studied trajectory. A summary of the consolidation units used in the cases can be seen in Table 26 below:

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pallet</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>4</td>
</tr>
<tr>
<td>Semi-trailer</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>4</td>
</tr>
<tr>
<td>ISO-container</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Non-standard</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

For the larger load units (ISO-containers, semi-trailers) the load unit had its own identity. This enabled an almost complete encapsulation of the Consignment where only a select few of its attributes were visible. The unit load was in these cases controlled as an individual by all TCSs, whereas the contained Consignment objects became hidden. In two of the cases (cases 2, 3), attributes of the Consignments were required to be visible even though a large unit load was used. This was because the consignments contained dangerous goods, and that their TCSs needed to have this information to comply with regulations.

Heterogeneous goods
It is described in Chapter 3.1.4 that when one or more of the parameters D, W, H, or L\(^\text{18}\) are outside their accepted ranges for a consignment the goods are called heterogeneous. It is also stated that heterogeneous goods cause higher cost than homogeneous goods. These costs are the result of state transitions that the goods require in order to achieve their goal state. By definition, the cost of a state transition for heterogeneous goods is calculated differently compared to the homogeneous goods in the same situation.

\(^\text{18}\) D = Density; W = Stowability; H = Ease of Handling; L = Liability
In Table 27 below, the heterogeneous goods parameters found in the cases are accounted for together with the control strategy used by the TCS to accommodate the requirements posed by the goods.

Table 27 Heterogeneous goods found in the cases and the strategies used to control them

<table>
<thead>
<tr>
<th>Case</th>
<th>Heterogeneous goods parameter</th>
<th>Strategy used to control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L: Liability. (Dangerous goods)</td>
<td>Separate trajectory inside terminal (L)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dangerous goods loaded separately onto semi-trailer (L).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special documentation produced (L).</td>
</tr>
<tr>
<td>2</td>
<td>S: Stowability, H: Hard to handle (all shapes and sizes are possible)</td>
<td>Stowed into ISO-container to improve S and H.</td>
</tr>
<tr>
<td></td>
<td>L: Liability (dangerous goods)</td>
<td>Markings on container (L)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special documentation (L)</td>
</tr>
<tr>
<td>3</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>4</td>
<td>S: Stowability, H: Hard to handle (large rolls)</td>
<td>Stowed into semi-trailer to improve S and H.</td>
</tr>
<tr>
<td>5</td>
<td>S: Stowability, H: Hard to handle (large rolls)</td>
<td>Stowed into semi-trailer to improve S and H.</td>
</tr>
<tr>
<td>6</td>
<td>L: Liability (a pallet may effectively disappear if any of the numerous operations during the filing process is erroneous)</td>
<td>Constant auditing of the card-file system to correct errors</td>
</tr>
<tr>
<td>7</td>
<td>H: Hard to handle (every box is unique and not loaded on pallet and must be sorted separately)</td>
<td>Sorting grid (simple rules) together with conveyor belt is used to improve H.</td>
</tr>
<tr>
<td></td>
<td>L: Liability (goods are theft-prone)</td>
<td>Security cage in terminal (L)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deliver instructions include rigorous safety aspects (L)</td>
</tr>
<tr>
<td>8</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
None of the studied cases displayed heterogeneous goods where density (D) was the cause. This does not automatically mean that the density parameter is not used when identifying heterogeneous goods. In the forwarding organisations that were studied, especially the one in cases 1 and 7, density is very much a factor that affects planning and control. The load planning system that the forwarder uses depends largely on the length the cargo requires, measured in meters. Extremely high-density goods may cause the length-calculations to fail since the density that is normally assumed (333 kg/m$^3$) also implies a certain standard volume. In the case of high-density goods, the semi-trailer may be fully loaded by weight even if it is only partly filled by length thereby, disrupting the planning.

The stowability (W) was identified as causing heterogeneity in cases 2, 4, and 5. The strategy was in all of these cases to encapsulate the goods into a container or semi-trailer so that the stowability attribute did not affect the further control of the consignment. In essence, the encapsulation effectively eliminates the stowability problem and replaces the various non-standard shapes and sizes of the goods with a standardised, stowable shape. There is a price to pay for this elimination, however, since the difficulty regarding stowability inevitably causes the semi-trailer/ISO-container to be less than 100% full.

Whenever goods are hard to handle (H), as was observed in cases 2, 4, 5, and 7, extensive, and often manual, labour is needed. Just as with the stowability, the handling is made easier if the consignment presents a standardised interface. In cases 2 and 7, manual labour was frequent, whereas cases 4 and 5 required specialised handling equipment that was only available at the consignor and the consignee. Encapsulation was the preferred strategy in cases 2, 4, and 5. By placing the hard-to-handle goods in an easy-to-handle load unit, the heterogeneity disappears. In case 7, the handling difficulty mainly depended on lack of pallets as well as lack of information. The only solution was to return order to the system by manually assessing the information coupled to each single box and to, gradually, consolidate parts of consignment into complete units. The handling is facilitated by a mechanical aid (the conveyor) and by a rule-of-thumb system (the sorting grid).

The liability aspect of heterogeneous goods (L) differs from the other three. This parameter is not necessarily physical, and may not be visible during a transportation process. The control strategies needed due to liability concerns vary greatly from case to case. In cases 1 and 2, dangerous goods caused several disruptions in the trajectories that were observed. In case 2, the encapsulation, which was made due to the stowability and handling difficulties, was rendered incomplete due to the fact that dangerous goods require markings on the exterior of the container. The consignor in case 7 demanded strict security measures of the TCS including a dedicated, caged area of the terminal facility. When examining the physical attributes of the consignments in case 7, they are not causing heterogeneity (apart from causing handling difficulty due to absence of pallets). The heterogeneity resides in the nature of the product being computers and therefore prone to theft. The liability aspect may take on many forms and a single, effective, control strategy is not possible to deduce. However, liability seems mostly to be related to the uncertainty-based complexity. The more knowledge about the states of a system, the easier it is to control aspects of liability factors.

**Heterogeneous goods increase complexity**

When a new consignment is added to a system that is under the control of a TCS, the TCS makes an assessment of the interface of the consignment as well as of the initial state and the goal state. If the goods are deemed heterogeneous, the requisite variety of the TCS increases. The capacity of the TCS, i.e. how many consignments it can regulate at the same time, depends largely on how many *interface types* (classes) the consignments can be divided into. By encapsulating consignments into subsystems such as unit loads the interfaces of the individual consignments are hidden inside the unit load. The TCS then regulates the trajectory
of the unit load and thereby of the contained consignments. As soon as a consignment contains heterogeneous goods, however, encapsulation with other consignments (homogeneous goods) may not be possible. Since the consignment contains heterogeneous goods, it will be handled individually more often than other consignments belonging to the homogeneous goods category, thereby increasing the state space.

By definition, the subsystems are autonomous until such a time as when the TCS needs to “step into” them in order to handle entropy. This does not mean that the TCS is not privy to the inner workings of the subsystems. When interviewed, an operator of a TCS may very well account for processes that take place within a subsystem. These accounts are not always correct, though, which leads to the conclusion that although the TCS may be the architect of the system, the subsystems encapsulate at least parts of their processes to the outside view. This discrepancy was the cause of problems in some of the studied cases, such as the load securing problem in case 4 or the introductory case described in Chapter 1.1.

Heterogeneous goods may require different trajectories than consignments containing homogeneous goods. The attributes that make the goods heterogeneous are also the same attributes that have to be manipulated or interpreted by the TCS in order to achieve the goal state. Therefore, heterogeneous goods are likely to require more data processing by the TCS compared to homogeneous goods. As seen in cases 2 and 3, the transportation of certain goods (mainly dangerous goods) require that several documents are created in advance and sent to various actors along the chain in preparation for the coming transport. For dangerous goods, several of the consignment properties (i.e. their outbound interface) must be known beforehand to be able to accurately plan the use of resources. In case 7, the arrival of semitrailers loaded with computers was often advertised by fax 24 hours before. The TCS used this interval to acquire extra personnel depending on the expected volume.

Evidence of uncertainty-based complexity has been found throughout the studied cases. One of the drivers behind this type of complexity is the number of decisions that cannot be made in advance (while planning the trajectory). Heterogeneous goods add to the uncertainty in the system by their very nature, since their interfaces contain parameters or parameter values that are considered outside normal ranges. Therefore, a TCS encountering heterogeneous goods in a consignment must adapt to the change in interface that this consignment presents in order to contain the uncertainty to the planning stage (as in case 2 where every transport was assumed to contain dangerous goods). If this uncertainty is allowed to spread beyond the planning stage, other parts of the trajectory may be affected by it. In these cases, decisions need to be made during the trajectory, and the encapsulated subsystems that have been employed to reduce the variety may experience difficulties in handling the uncertainty. In these situations, the TCS may have to step “into” the subsystems, thereby breaking the encapsulation in order to facilitate the consignment’s trajectory to the goal state. An alternative to this course of action is to employ subsystems that will handle the uncertainty on behalf of the TCS. This does not take care of the basic problem, though, as uncertainty still will exist on the TCS-level even if a Black Box is inserted into the system to handle it on lower levels.

Therefore, it can be stated that heterogeneous goods increase complexity in the transportation system. By definition, heterogeneous goods normally increase the uncertainty-based complexity, thereby causing the TCS to make decisions during the transportation process instead of when designing or planning the trajectory.

### 6.3.2 The interface of the TCS

There are several identified TCSs in the cases. In some cases there are multiple TCSs hierarchically organised (cases 2, 3, 4, and 5). Some of the cases had only one identified TCS
(cases 6 and 7). In case 1 and 8 there were more than one identified TCS, but on the same level (i.e. not hierarchically organised).

Table 28 The differences and similarities between the cases regarding TCS structure.

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than one TCS studied</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Hierarchically organised TCSs</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No discernable TCS hierarchy</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Only one TCS studied</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Verbal communication crucial</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The TCSs rely heavily on verbal communication. In almost all of the cases (except cases 6 and 7) the studied trajectories were depending on verbal communication between the TCS and other actors. Bookings and notifications were often verbal, even if a document also was sent. In case 5 for instance, the Main forwarder frequently calls the other actors in the case, even though documents are sent as well.

Hierarchically organised TCSs that have been studied encapsulate more than they know. A TCS on a high level has, as seen in several of the cases when creating the diagrams, little or no knowledge of the trajectories within the control of a TCS on lower levels. The high-level TCS can often describe the operations of the low-level TCS in great detail. In several of the cases, however, this description has proven wrong when studying the low-level TCS directly, which has been observed in cases 2, 3, 4, and 5. This can of course be explained as coincidence and that it is based on a sample that is too small to draw any conclusions. During the construction of the object models, however, all of the hierarchically organised cases displayed this property. If it is true – that the degree of encapsulation is higher than the actors are aware of – this may be an indication that even though the TCS knows its own interface, it does not know how this interface is perceived by others.

6.4 Visualisation during design – a calculation example

This section contains a calculation example of a terminal facility – based on the terminal in case 1 – where various types of goods are handled. The calculation example serves to demonstrate one of the uses of object-oriented design, where a model can be instantly tested and modified to test alternative designs. As was observed in the previous sections, in the long-term (and in some cases, medium-term) control scope the central complexity reducing strategy is to design a system that does not require much planning or creation of order to work.

The example is visualised in an object model and, through that model, a discrete-event system simulation environment (Witness)\(^{19}\), which has been used to perform the calculations. Witness is a discrete-event system simulation program operating in an object-oriented manner.

The terminal handles consignments of five goods types. Through a series of 16 calculations, the terminal is studied concerning its efficiency. As an addition to the cases in the previous

chapter, this can be seen as a visualisation of the terminal described in case 1. Much of the data is taken from case 1, such as number of operators, handling time etc.

In a calculated environment, any attribute can be set by the system designer. It is therefore of some interest to examine how a case previously studied can be altered to increase efficiency.

6.4.1 The calculated system

The system that is visualised can be idealised to an input-output system using the general transportation system model presented in Figure 33 on page 78.

The visualisation model consists of a terminal, where goods arrive in one end, are sorted, moved to the specified gate in the other end, and are loaded onto a vehicle. The system is designed for a certain number of consignments per day. If the number of consignments increases, or if the internal processing of the consignments takes longer, the system will not be able to process every consignment. In those cases, the terminal will not be emptied when the working day is over, i.e. there will be Work In Progress (WIP) left in the terminal. WIP is defined as the difference between input and output, i.e. the number of consignments left in the terminal.

![WIP diagram](image)

*Figure 60 WIP is measured as Input – Output.*

In real life, the terminal that the visualisation model is modelled from (case 1) does of course not have WIP at the end of a shift. The terminal frequently uses call-in personnel to be able to handle peaks. The simulation model is a simplification of the real system and is used to test whether certain attributes may affect the system efficiency. To calculate this, the number of resources must be controlled between experiments. With this measurement, the following applies:

- The more efficient the system is, the lower the WIP will be
  - The total complexity is assumed to decrease
- The longer each consignment takes to process, the higher the WIP will be
  - The total complexity is assumed to increase

The result variable WIP therefore reflects the complexity in the system.

6.4.2 Object model

The class diagram below represents the calculated system that is based on the collecting terminal in case 1. Note the manual coordination function that exists in the real-world counterpart to the model. This coordination function is an encapsulation that collects all the sorting decisions to one subsystem.
Figure 61 Class diagram of the calculated system.
The statechart is the same that was used in Case 1 (see Diagram 13 in Appendix 5). The following sequence diagram depicts the terminal process in the calculation:

![Sequence Diagram of the Calculated System](image)

**Figure 62 Sequence diagram of the calculated system**

### 6.4.3 The process

When a consignment arrives at an incoming gate, the goods are unloaded onto the loading area in connection to the gate. Goods of type B, dangerous goods, are placed in a separate area. The consignment is coordinated, i.e. it is designated with an outbound gate. This process is performed by the coordinator.

### 6.4.4 Goods types

Each goods type that has been used in the model displays differences in two dimensions. First, each goods type may require different amounts of time for each of the four processes (unloading, coordination, handling, and loading). Second, when handling the goods types, there may be different amounts of time needed to switch between two types.
The goods types are:

*Table 29 The five goods types used in the calculation model*

<table>
<thead>
<tr>
<th>Goods type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Pallet</td>
<td>A pallet containing general cargo</td>
</tr>
<tr>
<td>B Pallet w/ DG</td>
<td>A pallet containing general cargo, classed as dangerous goods</td>
</tr>
<tr>
<td>C Pallet, unsorted</td>
<td>A pallet containing consignments bound for more than one consignee</td>
</tr>
<tr>
<td>D Not on pallet</td>
<td>Goods not on pallet, requiring manual handling</td>
</tr>
<tr>
<td>E Pallet w/ monitoring</td>
<td>A pallet containing general cargo that for some reason needs individual attention</td>
</tr>
</tbody>
</table>

The assumptions are shown below (all assumptions are accounted for in Appendix 6).

*Table 30 The assumed space of time for the different goods types. Time in minutes.*

<table>
<thead>
<tr>
<th>Goods type</th>
<th>Unloading</th>
<th>Coordination</th>
<th>Handling</th>
<th>Loading</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg +/-</td>
<td>Avg +/-</td>
<td>Avg +/-</td>
<td>Avg +/-</td>
</tr>
<tr>
<td>A Pallet</td>
<td>1.0 0.3</td>
<td>0.7 0.5</td>
<td>1.5 0.4</td>
<td>1.3 0.2</td>
</tr>
<tr>
<td>B Pallet w/ DG</td>
<td>1.2 0.3</td>
<td>0.9 0.5</td>
<td>1.7 0.4</td>
<td>1.8 0.2</td>
</tr>
<tr>
<td>C Pallet, unsorted</td>
<td>1.0 0.3</td>
<td>2.0 0.5</td>
<td>1.5 0.4</td>
<td>1.3 0.2</td>
</tr>
<tr>
<td>D Not on pallet</td>
<td>1.5 0.5</td>
<td>0.7 0.5</td>
<td>5 2</td>
<td>2 1</td>
</tr>
<tr>
<td>E Pallet w/ monitoring</td>
<td>1.1 0.3</td>
<td>1.0 0.5</td>
<td>2.5 0.6</td>
<td>1.8 0.4</td>
</tr>
</tbody>
</table>

When a terminal operator switches between two goods types, the switch itself may take some time depending on the need for special equipment etc.
The table below contains the assumptions for the switching time:

*Table 31 The time for switching between goods types in minutes.*

<table>
<thead>
<tr>
<th>To\from goods type</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0.5</td>
</tr>
<tr>
<td>C</td>
<td>0.5</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1.5</td>
</tr>
</tbody>
</table>

6.4.5 Calculation results

16 calculations have been performed. The strategy has been to calculate the WIP when more than one goods type is allowed in the terminal. First, five calculations are performed, one for each goods type. This is done to find the goods type that renders the highest WIP without interaction with any of the others. Next, there are three calculations of 80, 60, and 20% each for goods type A, while the other types are distributed equally. Goods type A, which was studied in case study 1, is the normal goods type in the system. In the third round of calculations, goods type A represents 80%, whereas the distribution between the others varies. The goods types are removed from the calculation according to the results from the first five calculations, where the “easiest” goods type is removed with each new calculation. In the last four calculations, goods types A and the other four share the load 50-50.

The following diagram consists of the first five calculations:

*Table 32 Five calculations, one for each goods type*

<table>
<thead>
<tr>
<th>Calculation</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

The results are shown in the diagram below. In the visualisation model, some goods types take more time to process than others. The name of the goods type is at the bottom. The white part of each bar represents the number of consignments that the system was able to process.
The white parts represent the consignments that were left in the system at the end of each working day.

Figure 63 Results of calculations 1 to 5.

In the diagram above, it is evident that goods type D (“not on pallet”) affects the efficiency the most. Only 38% of the consignments complete the process in time. Not surprisingly, goods type A (“pallet”) affects efficiency the least with a WIP of only 2%.

The second round of calculations examines how the efficiency is affected when the other goods types increase in relation to goods type A. Three calculations are performed. In the first (calculation 6), goods type A is at 80% and the other types are at 5% each. Calculation 7 lowers goods type A to 60% which will put the other types at 10% each. In calculation 8, all five goods types are at 20% each.
Table 33 Three calculations where goods type A is in majority and the other four are equally distributed.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>80</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>60</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Figure 64 Results of calculations 6 to 8. The topmost partition in each bar consists of WIP left in the system each day.

As can be expected, the WIP-ratio increases as the ratio of other goods types, other than type A, increases. The next set of calculations keeps goods type A at a constant 80% (which is close to normal). The other goods types are then removed from the calculations one at a time according to their efficiency in calculations 1-5. Calculation 9 is the same ratio as calculation 6 (80-5-5-5-5). In calculation 10, goods type B is removed. In calculation 11, goods type C is removed and in calculation 12 only goods type D is left (except for goods type A).
Table 34 These calculations have goods type A at a constant 80% and let the other goods types share the remaining 20%. The other goods types are added so that the most time-consuming type (type D) is present in all four calculations.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>80</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>0</td>
<td>6.6</td>
<td>6.7</td>
<td>6.7</td>
</tr>
<tr>
<td>11</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 65 Results of calculation 9 to 12. The topmost partition of each bar consists of WIP left in the system at the end of each day.

Again, the effects of goods type D can be seen clearly above.

The last set of calculations examines the efficiency when goods types A and the other goods types share 50% of the consignments.
Table 35 In these calculations, goods type A has 50% and the others take turns filling up the other 50%.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 66 Results of calculation 13 to 16. The topmost partition is WIP at the end of each day.

Goods types B, C, and E do not have as much effect on the efficiency as goods type D. The efficiency reads a mere 56% for calculation 15.
6.4.6 Conclusions of calculation example

From the results of the calculations, the following can be deduced (about the model):

- Increase in number of interfaces lowers efficiency level somewhat. This parameter needs further examination, though. The switching time has a great influence on the effects of several goods types.

- Some of the goods types affect the system’s performance more than others. In the model, this is represented by somewhat longer time for the operations as well as for the switching between the goods types.

- The measurements are based on the assumption that goods are handled randomly. In a real-world system, in order to decrease switching time, goods of the same type may be handled together.

The four sets of calculations that were performed illustrate that the effects of goods types can be modelled by object-oriented methods and that such a model can be analysed using, for instance, simulation software. The results as such can in these calculations serve as a visualisation of how the effects of heterogeneous goods can be studied and what types of data are required for a successful simulation.

Although simulation software was used for this example, the results should not be attributed actual empirical value, since the data that was used for the simulation model partly was based upon assumptions.
6.5 Testing the research

This last part of the analysis regards the research itself.

According to Tashakkori and Teddlie (1998, pp.79-80), there are – in any research – two basic questions that have to be answered pertaining to collected data:

- Does the research measure/record what it intends to measure/record?
- Assuming that the research measures/records what is intended, is the measurement/recording without error?

The first question regards the validity and the second the reliability. There are four kinds of tests of research design (Yin, 1994, pp. 32-38; Riege, 2003):

Table 36 Four tests of research design and the tactics that can be used according to Riege (2003). The data in the table is modified by Riege from Yin (Yin, 1994, p. 33).

<table>
<thead>
<tr>
<th>Tests</th>
<th>Case study tactic</th>
<th>Phase of research in which tactic occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct validity</td>
<td>Use multiple sources of evidence</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Establish chain of evidence</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Have key informants review draft case report</td>
<td>Composition</td>
</tr>
<tr>
<td></td>
<td>Use within-case and cross-case analysis</td>
<td>Data analysis</td>
</tr>
<tr>
<td></td>
<td>Use visualisation tools such as diagrams to assist explanation building</td>
<td>Data analysis</td>
</tr>
<tr>
<td></td>
<td>Cross-check results to assure internal coherence of data</td>
<td>Data analysis</td>
</tr>
<tr>
<td>Internal validity</td>
<td>Use replication logic in multiple-case studies</td>
<td>Research design</td>
</tr>
<tr>
<td></td>
<td>Definition of scope and boundaries</td>
<td>Research design</td>
</tr>
<tr>
<td></td>
<td>Comparison of evidence with existing literature</td>
<td>Data analysis</td>
</tr>
<tr>
<td>External validity</td>
<td>Give full account of theories and ideas for each research phase</td>
<td>Research design</td>
</tr>
<tr>
<td></td>
<td>Assurance of congruence features between the research issues and features of the study design</td>
<td>Research design</td>
</tr>
<tr>
<td></td>
<td>Record observations and actions as concretely as possible</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Conduct pilot studies to test the case study protocol</td>
<td>Research design</td>
</tr>
<tr>
<td></td>
<td>Use a structured or semi-structured case study protocol</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Use multiple researchers</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Record data mechanically</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Develop case study database</td>
<td>Data analysis</td>
</tr>
<tr>
<td></td>
<td>Assure meaningful parallelism of findings across multiple data sources</td>
<td>Data analysis</td>
</tr>
<tr>
<td></td>
<td>Use peer review/examination</td>
<td>Composition</td>
</tr>
</tbody>
</table>
Each case study tactic listed in the table above will be addressed as they relate to this research in the following sections.

6.5.1 Construct validity

The construct validity represents how well the constructs, in this thesis the constructs of complexity drivers and heterogeneous goods, represent what is actually measured (Tashakkori and Teddlie, 1998, p 80). The evidence that has been collected may or may not support the constructs that were put to the test depending on the actual data as well as how the data is subjected to analysis. The construct validity can be addressed in two of the research phases: the data collection and the composition (the writing of this report).

In the data collection phase, multiple sources of evidence (also called triangulation) have been used whenever possible. Data triangulation is usually referred to in terms of cross-checking data to gain multiple sources of evidence for a phenomenon. However, multiple sources can also contribute to reveal new aspects of the studied systems (Dubois and Gadde, 2002; Hultén, 2002). By extensive note-taking during and after the interviews, cross-checking of several of the observed activities has been possible. This has given the possibility to find gaps as well as correlations in the data early, thereby establishing the chains of evidence towards validating the constructs.

The tactic to have key informants review draft case reports has been used, but not always. Since some of the studies have been directed towards the transportation of dangerous goods, there have been sensitive confidentiality issues. A mistake when transporting dangerous goods may well mean that a law is broken. Information like this is very sensitive and was given in the confidence that the identities of persons and organisations would be kept anonymous. Therefore, the case reports for instance, where a haulier admitted to breaking the law, could not be reviewed by the other actors in the same supply chain, because they know the haulier and will be able to identify him. In these cases, the data has been collected by two researchers at the same time, and notes have been compared afterwards.

6.5.2 Internal validity

Internal validity means that the collected data is analysed properly and that the right conclusions are made from the data. The data analysis phase is very important here, and the researcher needs to explore all the data as well as the relations that may exist between cases.

Within-case analysis is driven by the large amount of data in a case study and often consists of a simple chronological write-up of the case data (Eisenhardt, 1989). In this research, the within-case analyses are represented by the chapters 5.X.3 to 5.X.5 in this thesis (X denotes the case number).

The use of visualisation tools is evident when using the object-oriented models in Appendix 5.

The cross-checking of results is done extensively in Chapter 6.2. According to Eisenhardt, the cross-case analysis is used to counter the effects of bad interview data. People are poor information processors and will, willingly or not, produce faulty data to some degree. By doing cross-case pattern matching, these anomalies can be found (Eisenhardt, 1989).

6.5.3 External validity

The external validity measures how well the selected cases cover the area of study. It is important to bear in mind that already when designing the research, even though constructs and case data are valid, the context in which they are placed may be larger than the area that they cover. Therefore, it is essential that the research is designed in such a way that no key
areas are missing when selecting the cases, or that no relevant area of theory is missing when performing the analysis. Or, as Riege states:

“External validity is concerned with the extrapolation of particular research findings beyond the immediate form of inquiry to the general.”

(Riege, 2003, p. 81)

Eisenhart advocates that a case should replicate some part or property of other cases so as to enable cross-case comparisons and analyses (Eisenhardt, 1989). This replication logic has been used when selecting the cases for this thesis, as can be seen in Figure 40 on page 90, where various key segments of a model trajectory are seen in several cases.

When designing the research, the definition of scope and boundaries is crucial to the external validity. To be able to make any form of generalisation, the scope must be defined in such a way that a case sampling that represents the entire scope in some regard can be made. The scope in this thesis is mainly defined in Chapters 1.2 and 3.1.

The third tactic, the comparison of evidence with existing literature, is made implicitly, since a literature study on the uses of object-oriented methods in logistics and transportation is made in Chapter 3.5.

6.5.4 Reliability

The reliability means that the measurements that are used are accurate. As Yin states:

“The objective is to be sure that, if a later investigator followed exactly the same procedures as described by an earlier investigator and conducted the same case study all over again, the later investigator should arrive at the same findings and conclusions.”

(Yin, 1994, p. 36)

Therefore, it is important to document procedures that are followed in the case studies so that the process can be duplicated if need be.

According to the Table 36, one test of reliability is that the researcher gives a full account of theories and ideas for each research phase. The motive behind this is to put the reader into the frame or reference of the researcher at every stage of the research. This research project has a history that is briefly described in Chapter 3.1.4. The theories and presumptions that have guided the work have always been those belonging to the systems perspective. Early in the research the notion of state vs. transition arose, first addressed using risk analysis theory, later by cybernetics in combination with object-oriented methods. Most of the data gathering for the cases (cases 1-6, and parts of 7) was conducted during the “risk-analysis” phase. This data has subsequently been re-analysed (in this thesis) together with the new data from cases 7 (parts of) and 8. Even though risk-analysis has a history in the project, it is not accounted for to any lengths in this document. For a more detailed description of this phase the reader is referred to Arnäs (1999a).

Assurance of congruence features between the research issues and features of the study design means that when designing the studies, the research questions (applicable to this thesis) must match the case study design. The results from this research are partly grounded theory, partly deducted from applying existing theory to already gathered data. The research questions have been rewritten several times during the project – as they should. The preconceptions that were held in the beginning of the research have evolved together with the collected data as well as with the assimilated theory. Therefore, in this research project the design of each case study was relevant to the problem as it was stated at that time. In the beginning of each case description in Chapter 5, there is a section called Case objectives
(Chapter 5.X.1, where X denotes the case number) and Case methodology (Chapter 5.X.2). These sections describe the background for each case.

To record observations and actions as concretely as possible as well as to record data mechanically has proven to be a problem. Whenever possible, physical documents such as waybills etc. have been collected. But, as has been stated earlier, there have been no recordings of the interviews. This is due to the aforementioned fact that the cases involved dangerous goods and a tape recorder would probably not lead to truthful replies regarding mistakes that have been made. Whenever possible a fellow researcher has been present during interviews, as has also been mentioned before.

Case 1 was used as a pilot study to test the data collection method. As was mentioned earlier, the data has been re-analysed for this thesis which means that the analysis model that was used when the case study was conducted has since changed.

The use of a structured (or semi-structured) case study protocol has been the case from the beginning, and for the data collection specific guidelines have been developed (see Chapter 2.3.1). As was mentioned above, the model of analysis has been reworked during the project, thus changing the way the collected data has been used.

Whenever possible, multiple researchers (at least 2) have been used to perform interviews. The other researchers were either fellow PhD-students or senior researchers. In one case (case 7), some of the data comes from a student project. This data has been verified through interviews and first-hand observation. Cases 4, 5, and 8 were, due to limited funding, performed by the author alone. The data in these cases were verified by the interview objects.

The development of a case study database has taken several forms during the course of the project. In one stage an actual database application for registering case data was developed (Arnäs, 1999b). This was later abandoned due to difficulties in maintaining the application as such. The database now consists of several electronic documents, from the first impression notes to processed data, all searchable and sorted according to case number. The diagrams that are used in this thesis are also stored with the cases. Several of the collected documents (waybills etc.) have been scanned so that they can be examined on a computer.

The assurance of meaningful parallelism of findings across multiple data sources means that if the same findings exist in more than one case, the reliability of these findings is high. Chapter 6.1.2 contains such parallelisms, namely the class library that contains classes identified in all (or most) of the cases.

Previous work of the author has been subjected to peer review (Arnäs, 1999a; Arnäs, 1999b; Arnäs, 2001b; Arnäs et al., 1998; Arnäs and Sjöstedt, 1999). Parts of these publications are integrated into this thesis.

6.5.5 Summary of research testing

Several of the tactics lined up in Table 36 have been adopted in this research. Below is a summary of the tactics as they are adopted in this thesis:
Table 37 Case study tactics adopted in this thesis.

<table>
<thead>
<tr>
<th>Tests</th>
<th>Case study tactic</th>
<th>In this thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct validity</td>
<td>Use multiple sources of evidence</td>
<td>Used whenever possible</td>
</tr>
<tr>
<td></td>
<td>Establish chain of evidence</td>
<td>Possible due to multiple sourcing</td>
</tr>
<tr>
<td></td>
<td>Have key informants review draft case report</td>
<td>Used whenever possible</td>
</tr>
<tr>
<td>Internal validity</td>
<td>Use within-case and cross-case analysis</td>
<td>Used in all of the cases. See chapters 5.X.3 to 5.X.5 (X denotes the case number)</td>
</tr>
<tr>
<td></td>
<td>Use visualisation tools such as diagrams to assist explanation building</td>
<td>Used when building OO-based models. See Appendix 5</td>
</tr>
<tr>
<td></td>
<td>Cross-check results to assure internal coherence of data</td>
<td>Done in Chapter 6.2</td>
</tr>
<tr>
<td>External validity</td>
<td>Use replication logic in multiple-case studies</td>
<td>Has been considered when selecting cases. See Figure 40 on page 90.</td>
</tr>
<tr>
<td></td>
<td>Definition of scope and boundaries</td>
<td>The scope in this thesis is mainly defined in Chapters 1.2 and 3.1</td>
</tr>
<tr>
<td></td>
<td>Comparison of evidence with existing literature</td>
<td>A literature study on the uses of object-oriented methods in logistics and transportation is made in Chapter 3.5</td>
</tr>
<tr>
<td>Reliability</td>
<td>Give full account of theories and ideas for each research phase</td>
<td>Described in Chapter 3.1.4 and in Arnäs (1999a)</td>
</tr>
<tr>
<td></td>
<td>Assurance of congruence features between the research issues and features of the study design</td>
<td>See Case objectives (Chapter 5.X.1) and Case methodology (Chapter 5.X.2)</td>
</tr>
<tr>
<td></td>
<td>Record observations and actions as concretely as possible</td>
<td>No recordings were made (see Chapter 2.3.1)</td>
</tr>
<tr>
<td></td>
<td>Conduct pilot studies to test the case study protocol</td>
<td>Case 1 was used as a pilot study</td>
</tr>
<tr>
<td></td>
<td>Use a structured or semi-structured case study protocol</td>
<td>See Chapter 2.3.1 for data collection guidelines</td>
</tr>
<tr>
<td></td>
<td>Use multiple researchers</td>
<td>Used whenever possible</td>
</tr>
<tr>
<td></td>
<td>Record data mechanically</td>
<td>No recordings were made (see Chapter 2.3.1)</td>
</tr>
<tr>
<td></td>
<td>Develop case study database</td>
<td>All case data searchable on computer</td>
</tr>
<tr>
<td></td>
<td>Assure meaningful parallelism of findings across multiple data sources</td>
<td>See class library in Chapter 6.1.2</td>
</tr>
<tr>
<td></td>
<td>Use peer review/examination</td>
<td>Previous work of the author has been peer reviewed and partly integrated into the thesis.</td>
</tr>
</tbody>
</table>
7. CONCLUSIONS

This chapter concludes the thesis. There are three separate sections, the first containing guidelines for the design of interfaces, the second containing an account of the uses of the object-oriented framework in transportation and logistics and the third containing some suggestions for future research.

7.1 Designing interfaces
This section addresses the third research question:

Q3 How should the interface between the consignment and its TCS be designed in order to minimise the impact of complexity in the control of the transportation system?

In the previous chapter, the various aspects of control were examined. The long-term control mainly consists of design activities, where a TCS, in advance and long before any consignments arrive, defines the outer limits of the transportation system. The long-term control has the power to reduce not only the descriptive complexity, but also the computational and uncertainty-based. The descriptive complexity is the easiest to reduce in the long-term control scope.

7.1.1 Designing to reduce descriptive complexity

As seen in the cases, the manner in which the studied companies handled the descriptive complexity was through encapsulation. By creating a hierarchic model of nested, interacting, black boxes, or objects, the TCS may “delegate variety” from the regulated system to one of these objects, thereby constructing a capsule that reduces the variety to a few inputs and outputs. When designing to reduce descriptive complexity, the modeller will probably gain benefits from applying encapsulation whenever variety is likely to be large.

The key insight that is needed here is that a TCS does not have to do all the work itself. The requisite variety for a TCS that by itself controls an intermodal, international transport of a consolidated container is extremely large, to say the least.

When designing a system using the principles of OO, encapsulation becomes part of the design method when constructing class and use case diagrams. This does not mean that the descriptive complexity is minimised, however. The encapsulations that are made must be constructed in such a way that the “right” variety is encapsulated in the same subsystem. As was argued in Chapter 4.3.1, there are two types of encapsulation:

- Encapsulation of content. Limits outbound interface (number of attributes)
- Encapsulation of function. Limits inbound interface (number of operations)

Both types need to be synchronised in order to present a TCS with a canonical model of the encapsulated system. A system where more attributes are displayed than can be altered by operations is just as bad as a system where the results of operations are not displayed.
Therefore, the designer must carefully balance the widths of the inbound and the outbound interfaces so that they are mutually supportive.

The encapsulation must also account for hierarchic matching between the TCS and consignment hierarchies. On the goods side, the consignments are encapsulated into unit loads that are part of a nested hierarchy of consolidation levels, such as box-pallet-container-rail wagon-RoRo-ferry. Each of these consolidation levels presents its own interface specifying what handling equipment that is needed or what information that can be accessed. On the TCS side, a matching encapsulation of functions lead to hierarchically nested subsystems, as described above. Each TCS level is equipped with the requisite variety (knowledge as well as capacity) to control the trajectory of consignments on the matching level of consolidation.

In Figure 67 below, the simplification strategy of matching hierarchies is shown:

![Figure 67: Simplification by encapsulation.](image)

The matching of hierarchies is made by necessity. A level 4 TCS in the figure above will find it very difficult to control a consignment on consolidation level 1. The TCS needs control over the right handling resources as well as over the information structure of the consolidation level. Therefore, it is likely that hierarchically organised TCSs are only compatible with goods on a matching consolidation level.
7.1.2 Designing to reduce computational complexity

With the state space in place (all classes are designed, and the encapsulations are made), it is time to design the valid trajectories. When designing to minimise the computational complexity, the work should start as high up in the TCS hierarchy as possible. This was shown in the cases, where the number of valid trajectories was significantly diminished by the transport initiator and not by the TCS.

The length and number of valid trajectories through the state space drives the computational complexity. To define a valid trajectory, the designer must assign constraints to attributes as well as to operations. As with encapsulation, there are two types of constraints (taken from Chapter 4.3.2):

- Transitory constraints
  - Limiting operations, i.e. movements, or the transformations. These constraints do not affect the possible states of an object, but rather the operations that are available.
  - Limits the inbound interface
- Stationary constraints
  - Limiting states. Constrains the number of possible states for an object. Independent of the transitory constraints.
  - Limits the outbound interface

As with the encapsulations, the transitory and stationary constraints need to be synchronised to form coherent trajectories. When building OO-based models, the sequence diagrams and statecharts are well suited to the design of trajectories, both trajectories in design and trajectories in operation.

The transport initiator chooses a trajectory (or a class of trajectories) that are valid for a specific transport. This choice, in turn, decides the constraints that needs to be applied, and is often coupled with the choice of TCS. The TCS can be seen as a “trajectory-supplier” that makes available transformation packages to a transport initiator.

The trajectories that a TCS presents are mostly made up of smaller trajectories presented by other TCSs that work as sub-contractors to the main TCS. In Figure 68 below, the perceived trajectory supplied by TCS 1 is in fact a hierarchy of shorter trajectories supplied by other TCSs.
Therefore, the method to reduce the computational complexity is to delegate the responsibility to a subsystem so that only a virtual trajectory needs to be considered.

### 7.1.3 Designing to reduce uncertainty-based complexity

The uncertainty-based complexity has its cause in the lack of knowledge by the TCS of a particular state in the controlled system. If a TCS does not possess adequate information about the interface of a certain consignment, i.e. not a canonical model, the consignment will be difficult to control:

- If the TCS has insufficient information about the inbound interface, the consignment may behave erratically to input
- If the TCS has insufficient information about the outbound interface, the consignment may be considered to be unresponsive

The main strategies to handle uncertainty that was found in the cases were:

1. Step into the system that displays uncertainty
2. Employ another TCS that is more able to handle the uncertainty instead

The first strategy is one of variety. It requires that the TCS possesses the same variety as the system it enters, otherwise it would not be able to control the system once entered.

The second strategy is structural. By creating a hierarchy of TCSs, uncertainty can be isolated to certain levels of abstraction. When designing a class diagram for a system, the first step is to find all the types of entities – classes – in the system. Once this is done, they are related to each other, either in a generalisation-specialisation relationship, or an aggregation-decomposition relationship (see the class library in Chapter 6.1.2 for examples of this). Each class can then be assigned its own attributes and operations, thereby “taking responsibility” for its own transformation of input (operations) to output (attributes). As with the encapsulation, i.e. the choosing of class boundaries, the relationship between classes is of great importance. By defining the relationships properly, the designer creates communication channels between classes that can be used during a transportation process to send messages (orders) between the various objects. Therefore, by adding a structural element to the design
process, a designer may also prescribe how to handle future uncertainty by showing the objects where to find other objects that can be accessed if needed.

### 7.1.4 Designing for heterogeneous goods

As was seen in Chapter 6.3.1, heterogeneous goods mainly affected the uncertainty-based complexity. When designing a transportation system for heterogeneous goods it is important to define which of the parameters D, W, H, or L are causing the heterogeneity. Since the design phase takes place during the long-term (or in some cases medium-term) control scope, the actual heterogeneity is not known. The system can be more or less prepared, however, and that preparation can be part of the design.

Heterogeneity in density (D) may require additional load units (semi-trailers) or maybe other resource types. By assuring that the TCS has these resources within reach during the transportation process also ensures that the heterogeneity can be handled, although at a cost. An example of this was found in case 4, where an empty semi-trailer always was stationed close to the consignor just in case a transport was needed.

Poor stowability (W) may require alternative load units or more time allocated to load/unload the goods. The load units can be made available in the same way as when handling the density above. The need for additional loading/unloading time may require schedules to be revised or, at worst, completely rewritten. To design for this is difficult, since overly generous schedules are inefficient by definition. One way to address this problem is to allow for additional resources to assist in the loading/unloading so that the process does not take too much time. In cases 4 and 5, the poor stowability was solved by loading parked semi-trailers that were picked up by a semi-trailer tractor when they had been loaded.

Goods that are difficult to handle (H) can be so for two reasons:

1. The goods can only be handled manually, i.e. no standardised handling interface exists
2. The goods have a standardised handling interface, but the equipment that is needed is not present

When type 1 is encountered, the system designer can allow for extra manpower to be called in short-term, of course at a cost. When designing for type 2, the only way to ensure that the heterogeneity can be overcome is to make handling resources of the correct type available. In case 7, a conveyor belt was added just for this reason.

When faced with possible liability (L), the system designer can only ensure that the knowledge in the system is adequate to meet the requirements of the goods. Often, when designing a hierarchy of TCSs, liability issues are “bounced upward” until they can be addressed by the right TCS. In case 3, for instance, the main forwarder handles every disturbance in the system, even if the responsibility is delegated to another TCS.

### 7.2 The uses of object-oriented modelling in transportation

A literature review shows that object-oriented modelling and analysis are used in the field of transportation and logistics, although with various motives as well as results. The number of published references is relatively low, indicating either that (1) the references that are found are part of the early explorations into the uses of object-oriented methods within the domain of transportation and logistics, or (2) that the uses of object-orientation within this domain are not applicable and therefore should be disregarded. This thesis states the former to be true – that the object-oriented framework has definite uses within this domain and that transportation and logistics research will benefit from further explorations into the area.
7.2.1 Theoretical implications

When an object-oriented perspective is taken on a transportation system, there are some theoretical implications that can be noted. The main theoretical implications of choosing object-orientation are:

- A systems perspective is adopted.
- The view of the transportation system becomes discrete. The notion of a transportation system as a continuum is not possible when adopting an object-oriented view. The system takes on the role of a state-machine, sometimes coupled with a regulator to form a cybernetic system with regulator feedback loops. A separation between states and the transition between states must be made.
- An object-oriented model is canonical on the abstraction level chosen by the observer. This means that the model is completely observable (all possible states are visible) as well as completely reachable (all possible states can be reached through available transitions).
- Classes are separated from objects. By separating the class hierarchy from the system models, the classes can be reused in other models. The reuse of classes ensures that the abstraction level always stays the same if the same classes are used.

By adhering to the principles and guidelines of object-orientation, the following effects can be observed:

- The focus is concentrated upon *interfaces*, on the macro levels as well as the micro levels
  - Object-orientation is inherently interface-centric and will, regardless of abstraction level, reduce any system to a collection of related interfaces
- The process of design and redesign of systems, such as supply chains, are more efficient because of the reusability of classes

7.2.2 The importance of visualisation

Visual tools have been used in several areas of science as well as in practice for a long time, and the object-oriented framework contains a collection of several powerful such tools. There are of course other visualisation techniques and tools that are not part of object-orientation that may be equally – or more – potent in presenting complex and complicated information. The diagrams used in object-orientation have one advantage, though; they are part of a well-defined, large framework encompassing several aspects of systems science such as states, relationships, transitions, sequencing, message passing etc.

One thing is certain – the more complex the system, the greater the demands are for good visualisation tools to assist in the abstractions that are needed in order to gain any measure of control or understanding.

7.2.3 Analysis of current systems

When using object-oriented analysis, OOA, the diagrams and other tools within the framework are used to gain understanding of a system. By constructing an object model, the researcher is forced through a comprehensive checklist containing not only the various entities within the studied system but also their relationships, functions and behaviours. The finished model can for instance be converted into a simulation model for further analysis. The
process of building the model itself is an important part of the analysis, as the researcher is forced to enter and exit the system on several abstraction levels, thereby gaining comprehensive knowledge of several system levels. When doing case studies in the transportation and logistics field, the building of an object model can contribute to the researcher’s knowledge in ways that make possible cross-case comparisons.

7.2.4 (Re-)design of systems

By constructing a future system as an object model, further redesign of the system becomes more efficient. The reuse of classes gets more and more efficient with time.

In today’s business environment, mergers between transportation companies are commonplace, and with them follow mergers of transportation networks. By using a formalised, standardised modelling language, different systems can “learn” each other’s language and processes. The very concept of networks implies standardised interfaces between subsystems so that goods may traverse links and nodes as needed, without having to take into account company-specific deviations.

7.2.5 Meeting the requirements for interface design

In Chapter 7.1, guidelines for designing interfaces to reduce complexity were presented. These guidelines were:

- To reduce descriptive complexity:
  - Encapsulate, i.e. limit interface widths
  - Construct matching hierarchies of goods and TCSs
  - Balance inbound and outbound interfaces so that an external system gains a canonical representation

- To reduce computational complexity:
  - Apply constraints, i.e. limit interface depths
  - Arrange trajectories into hierarchies

- To reduce uncertainty-based complexity:
  - Design knowledge into the system so that decisions can be made during the transportation process

When constructing a class diagram, the designer automatically creates encapsulations each time a class is drawn. Whether these encapsulations meet the requirements or not is another issue. In order to be able to construct matching hierarchies, the modeller must identify not only the various hierarchical levels of the TCSs, but also of the consignments. In this phase, a class library can be used to supply the model with already finished classes that qualify.

The construction of the statecharts and sequence diagrams requires that the designer not only is familiar with the class diagrams but also with the functional purpose of the system. To be able to design trajectories, the designer needs to know about the initial state as well as the end state, at least regarding some of the attributes. The more attributes of these states that are known, the more detailed trajectories and statecharts can be constructed in advance. If possible, the designer arranges the trajectories hierarchically so that a separate TCS is in control of each trajectory level.

By constructing the use case diagrams, the designer can identify processes that may require decisions during the actual transportation process. There is therefore an opportunity for the
designer to ensure that the system contains the knowledge to handle these situations as they occur.

By using the object-oriented framework when designing transportation systems, several aspects are included in the design process, such as:

- The structural aspect
- The functional aspect
- The process aspect

When used correctly, the object-oriented framework facilitates the design of transportation systems where complexity is reduced.

7.3 A short summary of the findings

Four main findings were made in the course of the research project. They have each been presented in various locations in the thesis. This section contains a brief account of the findings and their core properties.

A class library of the cases. This class library is a synthesis of the object models of all the cases and consists of diagrams that evolved during the modelling process, as is consistent with the object-oriented techniques.

Heterogeneous goods increase complexity in transportation systems. Often, the complexity caused by heterogeneous goods manifests itself as uncertainty-based, requiring decisions to be made during the transportation process.

The control of a transportation system can be divided into three control scopes, each corresponding strongly to one of three complexity types. Each control scope has a time span, from long-term through medium- to short-term. Long-term control is mainly focused on design and aims at reducing the descriptive complexity. Medium-term control mainly consists of planning to reduce the computational complexity. The short-term control focuses on operations and monitoring to reduce uncertainty-based complexity.

Object-orientation as a modelling method is well suited for analysis and design of transportation systems. When designing a transportation system, object-orientation can be used to reduce complexity and to embed mechanisms that contribute to further complexity reduction.

7.4 Future research

This thesis is based on three approaches: heterogeneous goods, cybernetics, and object-orientation. These approaches can be extended and built upon further. This section contains ideas and suggestions for future work and development.

Heterogeneous goods and their effects on transportation systems is a topic that becomes increasingly interesting. World-wide transportation networks, which are designed to quickly forward goods over large distances at low cost, are increasing in size. Their vastness implies an economy of scale logic that depends on standardised interfaces and a select few unit load types. Because these networks apply encapsulation as a universal technique when handling their large flows, heterogeneous goods become harder and harder to accommodate using their standard products. Some large industrial companies are leaning the other way. They do not want to buy a standardised product, but rather require “their” transporter to solve every problem or demand they have, without having to adapt schedules and trajectories to other customers of the transportation company. This development has led to the large forwarding companies having diversified into one standardised transportation network, and a separate company to take care of the “troublesome” customers that demand something extra on top of
the ordinary service level. These companies are focused on providing high service levels towards their customer, even if they have to buy transport capacity from competitors to do it. A concept such as heterogeneous goods will probably be received differently in these two companies, the standardised and the specialised. Therefore, it would be interesting to see how the current development in the international transportation business affects the possibilities of deviations from the unit load dimensions or from the standardised information infrastructure that is emerging.

The uses of object-oriented methods in transportation and logistics are gaining in popularity. Transportation and logistics are well suited for the object-oriented family of methods, both for analysis and design. The expansion of the networks may also be helped by adhering to a structured model of visualisation to assist in assimilating/merging different networks with each other. A thorough examination of how a modelling/simulation/design/analysis of a real-world TCS would be an interesting project that could span across several years, yielding invaluable case data in the process. Since this thesis has taken a systems perspective, also with a fairly passive and information-centric modelling style, an action research study into the uses of object-oriented methods within a TCS organisation would be interesting.
8. REFERENCES


Arnäs, P. O. (1999a) GRSA - Goods requiring special attention and their effects on intermodal transportation systems, *Department of Transportation and Logistics*, Chalmers University of Technology, Göteborg, Sweden


Arnäs, P. O. (2001a) Case studies of two intermodal transportation chains between Sweden and Germany, Report 111, Chalmers University of Technology, Göteborg

Arnäs, P. O. (2001b) GRSA - Goods Requiring Special Attention and how to handle them, Nofoma 2001, Reykjavik

Arnäs, P. O., Lumsden, K. and Stjernman, G. (1998) Adding dangerous goods to the multi-modal logistics system - a comparative approach, Presented, 8th World Conference on Transport Research, Antwerp, Belgium


Encyclopædia Britannica (2007) "logistics",


Part of the Principia Cybernetica Web at http://mirror.bikkel.org/PCP/DEFAUL.html, 2006-12-19


Riege, A. M. (2003) Validity and reliability tests in case study research: A literature review with "hands-on" applications for each research phase, Qualitative Market Research, 6, p. 75.


Sjöstedt, L. (2005) A conceptual framework for analysing policy maker's and industry roles and perspectives in the context of sustainable goods transportation In Barriers to sustainable


Woxenius, J. (1998) Development of small-scale intermodal freight transportation in a systems context, *Department of Transportation and Logistics*, Chalmers University of Technology, Göteborg


## APPENDIX 1 – LIST OF TERMS

This section contains a list of the more important terms, and their meaning, as they are used in this thesis.

<table>
<thead>
<tr>
<th>Term</th>
<th>Abbr.</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated Guided Vehicle</td>
<td>AGV</td>
<td>A mobile robot used to transport materials and products within a facility</td>
</tr>
<tr>
<td>Ashby’s Law</td>
<td></td>
<td>See Requisite Variety, the Law of</td>
</tr>
<tr>
<td>Attribute</td>
<td>$S_i$</td>
<td>A variable that is part of a system’s (or object’s) state. The state is the collective value of all the attributes. The attributes form a vector.</td>
</tr>
<tr>
<td>Bill of Lading</td>
<td>BoL</td>
<td>A document that is sent from consignor to consignee regarding a (sea) transport. The consignee uses the BoL as a requisition to gain control of the consignment from the port.</td>
</tr>
<tr>
<td>Canonical model</td>
<td></td>
<td>A model where every visible state is also reachable through visible operations. Also, all operations must result in visible states.</td>
</tr>
<tr>
<td>Class</td>
<td></td>
<td>In object-orientation, a class denotes a subsystem boundary that is used to group entities with the same interface structure together.</td>
</tr>
<tr>
<td>CMR</td>
<td>CMR</td>
<td>International waybill. Used by a forwarder when transporting across borders.</td>
</tr>
<tr>
<td>Consignee</td>
<td></td>
<td>Receiver of goods</td>
</tr>
<tr>
<td>Consignment</td>
<td></td>
<td>A consignment consists of the physical objects that from an administrative point of view are treated as a single shipment and therefore - throughout a specific part of a supply chain - are given a unique identity. Defined on page 6.</td>
</tr>
<tr>
<td>Consignor</td>
<td></td>
<td>Sender of goods.</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td>Limits the range of attributes or operations.</td>
</tr>
<tr>
<td>Container Packing Certificate</td>
<td>CPC</td>
<td>Issued by the person responsible for loading a container with dangerous goods. The CPC certifies that the container is loaded according to regulations.</td>
</tr>
<tr>
<td>Customised Transportation Document</td>
<td>CTD</td>
<td>In one of the cases, case 8, this denotes a consignee-specific CMR.</td>
</tr>
<tr>
<td>Cybernetics</td>
<td></td>
<td>“/.../ the entire field of control and communication theory, whether in the machine or in the animal” (Wiener, 1961)</td>
</tr>
<tr>
<td>Dangerous goods</td>
<td>DG</td>
<td>Products that are classified by the UN as dangerous for transportation.</td>
</tr>
<tr>
<td>Dangerous Goods Declaration</td>
<td>DGD</td>
<td>A document declaring the type, packaging and quantity of dangerous goods in a consignment (for sea transport).</td>
</tr>
<tr>
<td>Dangerous Goods fact sheet</td>
<td></td>
<td>A document declaring facts about a substance such as chemical properties and emergency procedures. Issued by the manufacturer of the substance and used as supplementary information when transporting the substance.</td>
</tr>
<tr>
<td>Term</td>
<td>Abbr.</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DG fact sheet</td>
<td></td>
<td>See Dangerous goods fact sheet</td>
</tr>
<tr>
<td>Disturbance potential</td>
<td>$D_p$</td>
<td>A two-dimensional measurement of combinations of probabilities and consequences for disturbances. The disturbances for a curve in an FN-diagram (from risk analysis). This curve represents the potential disturbances for a given situation.</td>
</tr>
<tr>
<td>Encapsulation</td>
<td></td>
<td>Hiding attributes and operations within a system, thus making the interface narrower.</td>
</tr>
<tr>
<td>Entropy</td>
<td></td>
<td>A measurement of the indeterminateness, or uncertainty, in a system.</td>
</tr>
<tr>
<td>Full Truck Load</td>
<td>FTL</td>
<td>“FTL: A term which defines a shipment which occupies at least one complete truck trailer, or allows for no other shippers goods to be carried at the same time.” (CSCMP, 2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defined on page 43.</td>
</tr>
<tr>
<td>Goal state</td>
<td></td>
<td>The desired state of a (regulated) system</td>
</tr>
<tr>
<td>Goods Requiring Normal Attention</td>
<td>GRNA</td>
<td>The opposite of GRSA</td>
</tr>
<tr>
<td>Goods Requiring Special Attention</td>
<td>GRSA</td>
<td>Defined as goods that, because of their disturbance potential, prompt activities in a transportation system.</td>
</tr>
<tr>
<td>Heterogeneous goods</td>
<td></td>
<td>Goods where one or more of the parameters Density (D), Stowability (S), Difficulty of handling (H), or Liability (L) are outside their accepted ranges. Defined on page 48.</td>
</tr>
<tr>
<td>Homogeneous goods</td>
<td></td>
<td>Goods where all of the parameters Density (D), Stowability (S), Difficulty of handling (H), or Liability (L) are inside their accepted ranges. Defined on page 48.</td>
</tr>
<tr>
<td>Inbound interface</td>
<td></td>
<td>The collection of visible inputs for a system. Defined on page 7.</td>
</tr>
<tr>
<td>Information entropy</td>
<td></td>
<td>See Entropy</td>
</tr>
<tr>
<td>Inheritance</td>
<td></td>
<td>A concept in object-orientation where the interface of a class is passed on to its hierarchical children</td>
</tr>
<tr>
<td>Intermodal</td>
<td></td>
<td>Transportation of goods using more than one mode of transport.</td>
</tr>
<tr>
<td>Less than Truck Load</td>
<td>LTL</td>
<td>Smaller shipments of freight utilising a network of terminals and relay points (CSCMP, 2006). Defined on page 43.</td>
</tr>
<tr>
<td>Merge-in-transit</td>
<td></td>
<td>When two flows of goods are consolidated into a single consignment en route between consignor and consignee</td>
</tr>
<tr>
<td>Modularity</td>
<td></td>
<td>In object-orientation, the notion of standardised interfaces that makes it possible to replace a component (subsystem) with another, as long as the interface is the same.</td>
</tr>
<tr>
<td>Object</td>
<td></td>
<td>A real-world entity belonging to a class</td>
</tr>
<tr>
<td>Object Model, the</td>
<td></td>
<td>See Object-Orientation</td>
</tr>
<tr>
<td>Object technology</td>
<td></td>
<td>See Object-Orientation</td>
</tr>
<tr>
<td>Term</td>
<td>Abbr.</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Object-Orientation</td>
<td>OO</td>
<td>A collection of methods for designing and analysing systems</td>
</tr>
<tr>
<td>Operation</td>
<td>O_i</td>
<td>Part of the inbound interface of a system. The operations form a vector. The only way to affect a system’s state is to execute one of its operations.</td>
</tr>
<tr>
<td>Outbound interface</td>
<td></td>
<td>The collection of visible outputs for a system. Defined on page 7.</td>
</tr>
<tr>
<td>Overpack</td>
<td></td>
<td>A plastic coating, often shrink-wrapped plastic, over a pallet so that any contents on the pallet are bound together.</td>
</tr>
<tr>
<td>Regulator</td>
<td></td>
<td>In cybernetics, a regulator controls the state of a regulated system</td>
</tr>
<tr>
<td>Requisite Variety, the Law of</td>
<td></td>
<td>A regulator must possess at least the same variety as the system it regulates</td>
</tr>
<tr>
<td>RFID tag</td>
<td></td>
<td>An RFID tag is a small object, such as an adhesive sticker, that can be attached to or incorporated into a product. RFID tags contain antennas to enable them to receive and respond to radio-frequency queries from an RFID transceiver (Wikipedia, 2007).</td>
</tr>
<tr>
<td>Sender certificate</td>
<td></td>
<td>For a road transport of dangerous goods, this document certifies that the consignor has packed and labelled the dangerous goods according to regulations</td>
</tr>
<tr>
<td>Shipping Advice</td>
<td></td>
<td>A list of arriving consignments (by sea) that is sent by the forwarder on the sending side to the forwarder on the receiving side</td>
</tr>
<tr>
<td>State</td>
<td></td>
<td>The collective value of an object’s attributes</td>
</tr>
<tr>
<td>State space</td>
<td></td>
<td>See Variety</td>
</tr>
<tr>
<td>Terminal</td>
<td></td>
<td>A terminal is a node in a network that bridges the gaps between link capacities, link frequencies, and link arrival times.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defined on page 42.</td>
</tr>
<tr>
<td>Terminal facility</td>
<td></td>
<td>A terminal facility is a physical structure that accommodates the needs of different vehicles in terms of loading equipment, storage areas, and information handling etc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defined on page 42.</td>
</tr>
<tr>
<td>Trajectory</td>
<td></td>
<td>The temporal evolution of a system takes the form of a curve, or trajectory, in the state space. Defined on page 4.</td>
</tr>
<tr>
<td>Transformation</td>
<td></td>
<td>A change of state in a system</td>
</tr>
<tr>
<td>Transport Control System</td>
<td>TCS</td>
<td>A TCS is a system that controls the trajectory of a transportation process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defined on page 5.</td>
</tr>
<tr>
<td>Variety</td>
<td></td>
<td>The number of states a system can assume</td>
</tr>
</tbody>
</table>
APPENDIX 2 – OBJECT-ORIENTATION – THE BASICS

This appendix contains a recapitulation of the basic concepts in object-orientation. It is intended for readers not familiar with object-orientation, or for those that wish to refresh their previous knowledge. The material presented here is taken from the following sources:


The above books are recommended for further reading on the applications of object-orientation and UML.

The object-oriented framework is a collection of several different modelling methods and visualisation techniques. Each method/technique has its own history and although they have been refined and “streamlined” as parts of the OO-framework, they are distinctly different and will be presented here as the basic concepts of OO. A more detailed description of the various diagram types can be found in Appendix 3 where the Unified Modeling Language is described.

A-2.1 Classes and objects

An object is defined by Booch as a tangible entity that exhibits some well-defined behaviour. The definition is expanded to:

“An object has a state, behaviour, and identity; the structure and behaviour of similar objects are defined in their common class…”

(Booch, 1991)

Thus, a class is a grouping of objects sharing the same data structure and behaviour. Examples of classes and containing objects are:

- Class: Vehicle
  - Object: Bicycle
  - Object: RoRo-ferry
  - Object: Station wagon

Just because an object belongs to a certain class, this does not mean that it cannot belong to other classes as well. The object Bicycle is for instance also a member of the classes Bicycles, Non-motorised transports, Two-wheeled vehicles etc.
Each of the defined classes represents traits that are common for all objects belonging to the class. For example, examine the classes below, all containing the object Lorry:

- Class: Vehicle
  - Important traits are how it can be driven and manipulated in traffic
- Class: Transportation resource
  - Important traits are position, load capacity etc
- Class: Product
  - Important traits are market value, production cost etc.

Each of the classes adequately describes the object Lorry, but from different viewpoints. The definitions of classes are thus very much the product of the modeller’s preconceptions and areas of interest. A lorry manufacturer would for instance probably use the Product class to describe a lorry, whereas a driver would use the Vehicle class.

So, a class depicts objects that share a *structure*. This structure is in object-orientation defined by two lists:

- List of attributes
- List of operations

The attributes are a number of (measurable) values that depict the *state* of the object. For the Vehicle class, attributes could for instance be:

- Position (GPS coordinates Lat, Long)
- Speed (in km/h)
- Destination (GPS coordinates Lat, Long)

The state for a certain vehicle could then be described thus:

- Position: Lat=57.693277; Long=11.975462
- Speed: 34 km/h
- Destination: Lat=57.362844; Long=12.090808

The collective value of the attributes is thus called the object’s *state*.

The operations contain calls to processes that objects, belonging to a class, can perform. Typical operations that can be listed in the Vehicle class include:

- Start
- Stop
- Drive to GPS position (Lat; Long)
In our example above, the following series of operations would then order the vehicle object to a new destination:

- Start
- Drive to (Lat=57.362844; Long=12.090808)
- Stop

A class is often depicted like this:

<table>
<thead>
<tr>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Position: Lat</td>
</tr>
<tr>
<td>+Position: Long</td>
</tr>
<tr>
<td>+Speed</td>
</tr>
<tr>
<td>+Destination: Lat</td>
</tr>
<tr>
<td>+Destination: Long</td>
</tr>
<tr>
<td>+Start()</td>
</tr>
<tr>
<td>+Stop()</td>
</tr>
<tr>
<td>+Drive to(Lat, in Long)</td>
</tr>
</tbody>
</table>

On top is the class name, next follow the attributes and at the bottom the operations.

### A-2.2 Relationships between classes

One of the many significant features of object-oriented modelling is the way in which class relationships can be described. There are two basic types of relationships between classes: aggregation-decomposition and generalisation-specialisation.

Aggregation-decomposition relationships describe whole-part relationships between classes. A few examples:

- The class Bicycle contains one, two, or three objects of the class Wheel
- The class Station wagon contains exactly four objects of the class Wheel

In a diagram, the two relationships look like this:

```
  Bicycle  Station wagon
    ↓        ↓
    1..3     1
   
  Wheel    Wheel
  ↓        ↓
  1        4
```

The notation means that one object of the Bicycle class consists of 1 to 3 objects of the Wheel class.
This notation can be expanded to include other objects:

The second relationship type is generalisation-specialisation, where classes are ordered to form a hierarchy based on specialisation of attributes and operations. For instance, a Vehicle may be either a Bicycle or a Station wagon. These relationships are drawn like this:

The arrow can be read as a top-down description: A Vehicle object can be a Station wagon or a Bicycle. Another way to interpret it is by using a bottom-up description: A Bicycle object is a kind of Vehicle.
These generalisation-specialisation hierarchies can contain several levels:

By combining these two relationship types, the following diagram can be constructed:

The diagram above contains much information about not only the class Vehicle, but also about all the other classes. This means that when an object of class Station wagon is studied, a number of other classes are included in the description because their relationships have been previously defined.
A-2.3 Inheritance

In the diagram above, there are no attributes or operations listed for the classes except for Vehicle. This does not mean that they do not exist, however. An important concept in object-orientation is that of inheritance. It means that an object inherits the attributes and operations from its parents. In our example with the Station wagon, it inherits all of the attributes and operations from Vehicle and they do not have to be written out explicitly. A subclass (like Bicycle or Station wagon) can of course contain attributes and operations of its own that are specific to them:

The diagram above shows that an object of the class Station wagon not only has all the attributes and operations of its superclass Vehicle, but also some of its own.
APPENDIX 3 – UML – UNIFIED MODELING LANGUAGE

This Appendix briefly describes some of the different diagrams used in the Unified Modeling Language, UML. UML is the creation of the Object Management Group (OMG).

“The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems.”

(Object Management Group, 2001, p. 1-1)

In terms of the views of a model, the UML defines the following graphical diagrams (Object Management Group, 2001, p. 1-4):

- use case diagram
- class diagram
- behaviour diagrams:
  - statechart diagram
  - activity diagram
  - interaction diagrams:
    - sequence diagram
    - collaboration diagram
- implementation diagrams:
  - component diagram
  - deployment diagram

A-3.1 The Use Case diagram

The first diagram to create when building a UML model is the Use case diagram. By doing this, the system is thoroughly studied. The Use case diagram (together with the Class diagram) is the starting point for the other diagrams within the UML specification. Identifying the different use cases in the studied system is the first step. A use case is a composite of actions that can be described with words as “Order products from consignor” or “Handle goods in terminal”. Identifying and naming each use case in the system leads to the next step: identifying the actors. Each actor plays a certain role. Therefore, two or more actors can in fact be one and the same person, but playing different roles.

"An actor defines a coherent set of roles that users of an entity can play when interacting with the entity. An actor may be considered to play a separate role with regard to each use case with which it communicates.”

(Object Management Group, 2001, p. 3-97)

By drawing the use cases within a system including the actors that interact and communicate with the cases, the Use case diagram is completed. In the example below, Actor2 is only involved in UseCase1, and Actor3 in UseCase3, whereas Actor1 is involved in all three Use cases.
Each use case can then be described in the Sequence diagram.

A-3.2 The Class diagram

The static structure of a system can be defined in a Class diagram. By identifying the different concepts in the problem domain (here named classes), a hierarchical diagram can be constructed based upon the classes’ relationship with each other. Each class is defined by its own attributes. Attributes can only change value when the class’ operations are executed. By viewing a system in this fashion, a focus on essential information is maintained. Each class needs a certain amount of data to be able to function (i.e. maintain attributes and execute operations). The amount of data needed can be seen as the difference between order (in the form of a functioning class) and chaos (the class degenerates due to lack of data). An example of a Class diagram is shown in Figure 70. Class 1 can be of two subtypes (Class 2 or Class 3). If an instance of Class 1 is of subtype Class 2, the attributes 1 to 7 are all valid (and the operations 1 to 7). It is said that Class 2 inherits the attributes and operations of Class 1. Class 1 also has an association to Class 4. In this case, each instance of Class 1 “contains” 0 or more instances of Class 4.
Each instance of a certain class is called an *object*. The symbol for an object looks like the class symbol. The name of the object is followed by the name of its class (the class acts like a template for the object). The name is underlined to prevent mix-up between classes and objects. Each object is unique, even if another object exists with the same values of its attributes.

The *state* of an object is the collective value of its attributes. To change states, one or more of the class’ operations need to be performed.
A-3.3 The Sequence diagram

In the sequence diagram, all objects within a certain use case are represented (the actor is also an object here). There is a time axis going downwards and objects communicate by sending messages. There are different kinds of messages:

- Normal message – open-ended arrow with solid line. Used to indicate that information is passed on to the receiving object (“I have arrived”).
- Function call – closed arrow with solid line. Interpreted as an operation request by the receiving object (“Open the gate!”).
- Return message – dotted line. Used as a response to a previous message (“Thank you for arriving. I am finished with you. You may now depart.”)
- Call to self – arrow going to the same object that it originates from. This can be a way of showing activities inside the specific object (“Load goods onto truck. Continue until finished”).

Each object has its own lifeline (the vertical dotted lines). An X at the end of the lifeline indicates that the object is destroyed after its last message. When an object is active, i.e. sends a message or processes information, the lifeline is thicker, forming a narrow box for the duration of the active period.

![Sequence diagram containing three objects and their interaction](image)

A-3.4 The Statechart diagram

Each object can, in a use case, assume several states. Each state is defined by the object’s attribute vector meeting some pre-defined conditions. The Statechart diagram represents these changes. The diagram always has a starting point (solid black circle). As explained earlier, the only way to change states is to perform one or more operations. In the diagram below, State1 is assumed when operation1 is executed. It is possible for an object in a certain state to perform a number of operations that do not change states (the way it is defined), see operation3. One example of these non state-changing operations is: “Rotate pallet so that it faces exit”. The way the state is defined (in this example let us assume “Pallet near exit”)

200
means that the state is still current even if an operation has been performed. It is possible to include conditions in the diagram, enabling a choice of operation based on, for instance, attribute values. The Statechart is ended with an end state symbol, a solid black circle with another black circle around it. The Statechart displays the possible trajectories in the given state space.

Figure 73 A Statechart diagram displaying 7 states (including start and end) and 8 operations, forming a model of all possible trajectories in the state space.
APPENDIX 4 – REVIEWED PAPERS

This Appendix contains a list of the papers that were reviewed in Chapter 3.5 on page 72. The table consists of a detail of the studied papers, their approaches and application areas. The number one (1) in for instance grid 1B means that the paper uses object-orientation in the design of manufacturing systems. The bottom row contains a sum for each combination of area and approach (table continued on next page).

The approaches are:
1. OO-modelling is used to design real-world systems
2. OO-modelling is used to build a control system
3. OO-modelling is used to build a simulation model
4. OO-modelling is used to analyse a system

There are five different application areas, A to E:
A. Transportation
B. Manufacturing
C. Automation
D. Public transport
E. Warehousing

<table>
<thead>
<tr>
<th>Reference</th>
<th>Approach 1</th>
<th>Approach 2</th>
<th>Approach 3</th>
<th>Approach 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfieri and Brandimarte (1997)</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Basnet et al. (1990)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Betlem and van Aggele (1994)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bullinger et al. (1997)</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Cielatkowski and Goczyla (1997)</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Du and Wu (2001)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ehlers and van Rensburg (1996)</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Foster (1998)</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Govindaraj et al. (2000)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grabot and Huguet (1996)</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>van der Heijden et al. (2002)</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Hung et al. (2004)</td>
<td>1 1</td>
<td></td>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>Kim and Rogers (2005)</td>
<td>1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kovacs et al. (1999)</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Lau and Lee (2000)</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Reference</td>
<td>Approach 1</td>
<td>Approach 2</td>
<td>Approach 3</td>
<td>Approach 4</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Mak et al. (1999)</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marín et al. (1998)</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McCormack and Roberts (1996)</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moayyed and Vosniakos (1998)</td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Ottjes and Duinkerken (1996)</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paolucci and Pesenti (1999)</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>Praehofer and Pichler (1995)</td>
<td>A</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Rowe et al. (1997)</td>
<td>A</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Schmidt (1996)</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shah et al. (1997)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Thrampoulidis et al. (1997)</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>Ton et al. (1998)</td>
<td>A</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Tran and Greewal (1997)</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Veeke and Ottjes (2000)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Wache (1998)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Wang (1996)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Wang (1999)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Yeung and Moore (1996)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Yun and Choi (1999)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Zhang et al. (1999)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Sum</td>
<td>16</td>
<td>16</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX 5 – OBJECT MODELS BASED ON THE CASES

A-5.1 Generic class diagrams – Class library

The following class diagrams are constructed based on all the cases. They show the various generic classes that are found in most of the studied cases. Subsequent models of the cases will implicitly refer to these basic, generic diagrams.

A-5.1.1 Class diagram for the relation TCS-Consignment

The Consignment class is an aggregated entity that consists of Load units and Information objects. It is created by an Actor. Its transport is facilitated by a TCS.

Diagram 1 Class diagram for the relation between TCS and Consignment

The two information types Electronic and Physical will have to be extended uniquely for each case.

A-5.1.2 Class diagram for the Load unit

A Load unit contains Products. It also has a Capacity. The Load unit can be of various types, such as Pallet, Container, Box, etc. Load units are recursive, i.e. they may contain other load units.
Diagram 2 Class diagram for a load unit

A-5.1.3 Class diagram for the relation TCS-Resource

A TCS controls several Resources. Each Resource can be of several different types. Resources are recursive, i.e. a Resource may contain other resources if possible, and for instance a Vehicle may contain a Tail lift or a Forklift.
Diagram 3 Class diagram for the relation between TCS and Resource

A-5.1.4 Class diagram for the Facility

A Facility can be a Loading dock a Terminal etc. It is operated by a Facility operator and contains Resources. It may also have an Address.
Diagram 4 Class diagram for the Facility

A-5.1.5 Class diagram for the Actor

An actor can be either a Person or a Company. The Actor can play several different Roles, such as Consignor, TCS or Consignee.

Diagram 5 Class diagram for the Actor

A-5.1.6 How the generic classes will be used

By defining these generic classes, modelling of the actual cases becomes easier. If, for instance a resource in a case is a Manual forklift, the class that is used will be just the Manual forklift, but with attributes and operations from its “parents” and “grandparents” embedded as well. This saves the modeller the effort of creating the Resource-hierarchy for every new case. Such a class library can also be extended freely as, for instance, new resource types are found. The addition of another type of fixed lifting equipment to the Resource diagram will not affect existing models already built based on the class library.
A-5.2 Case 1

A-5.2.1 Class diagrams

Diagram 6 The Consignment class in Case 1
Diagram 7 The terminals in Case 1

Diagram 8 A Vehicle in Case 1
A-5.2.2 Use cases

**Diagram 9 Use case for the Collecting terminal TCS in Case 1**

- Unloader of goods
- Coordinator
- Driver of trailer truck
- Driver of lorry
- Internal transporter
- Loader of goods

**Diagram 10 Use case for the Dispersing terminal TCS in Case 1**

- Unloader of goods
- Coordinator
- Driver of trailer truck
- Driver of lorry
- Internal transporter
- Loader of goods
A-5.2.3 Sequence diagrams

Diagram 11 Sequence diagram for the Collecting terminal TCS in Case 1

Diagram 12 Sequence diagram for the Dispersing terminal TCS in Case 1
A-5.2.4 Statecharts

Diagram 13 Statechart for the Consignment object in the Collecting terminal TCS in Case 1

Diagram 14 Statechart for the Waybill object in the Collecting terminal TCS
Diagram 15 Statechart for the Waybill object in the Dispersing terminal TCS in Case 1
A-5.3 Case 2

Diagram 16 Class diagram for the Transporter terminal in Case 2

Diagram 17 The various types of Consignment documentation in Case 2
A-5.3.2 Use case diagrams

Diagram 18 Various types of Container documentation in Case 2

Diagram 19 Use case diagram for the Forwarder as TCS in case 2

Diagram 20 Use case diagram for the Transporter as TCS in case 2
A-5.3.3 Sequence diagrams

Diagram 21 Sequence diagram for the sending side in Case 2

A-5.3.4 Statecharts

Diagram 22 Statechart for the Consignment object on the sending side in Case 2
A-5.4 Case 3

A-5.4.1 Class diagrams

Diagram 23 Class diagram for the different vehicles and load units used in Case 3
A-5.4.2 Use case diagrams

Diagram 24 Use case diagram for the Swedish forwarder as TCS in case 3
Diagram 25 Use case diagram for the Ferry operator as TCS in case 3

Diagram 26 Use case diagram for the Intermodal terminal operator 1 as TCS in case 3
Diagram 27 Use case diagram for the Local forwarder as TCS in case 3
Diagram 28 Sequence diagram for Case 3
Diagram 29 Sequence diagram for the transport from Intermodal terminal to consignee in Case 3

A-5.4.4 Statecharts

Diagram 30 Statechart for Case 3
A-5.5 Case 4

A-5.5.1 Class diagrams

Figure 74 Class diagram over the various transport bookings and their related documents in case 4
A-5.5.2 Use case diagrams

Figure 75 Class diagram of vehicles in case 4

Diagram 31 Use case diagram for the Main forwarder as TCS in case 4
Diagram 32 Use case diagram for the Swedish sub-forwarder as TCS in case 4

Diagram 33 Use case diagram for the Consignor as TCS in case 4
Diagram 34 Use case diagram for the Ferry operator as TCS in case 4
A-5.5.3 Sequence diagrams

Diagram 35 Sequence diagram 1 for Case 4
Diagram 36 Sequence diagram 2 for Case 4

A-5.5.4 Statecharts

Diagram 37 Statechart for the trailer in case 4
Diagram 38 Expansion of the state In Swedish ferry terminal in case 4

Diagram 39 Expansion of the state In German ferry terminal in case 4
A-5.6 Case 5

Diagram 40 76 Class diagram over the various transport bookings and their related documents in case 5
A-5.6.2 Use case diagrams
Diagram 43 Use case diagram for the Intermodal operator as TCS in case 5

Diagram 44 Use case diagram for the Swedish rail terminal operator as TCS in case 5
Diagram 45 Use case diagram for the Consignor as TCS in case 5
A-5.6.3 Sequence diagrams

Diagram 46 Sequence diagram for the first part of case 5
Diagram 47 Sequence diagram for the second part of case 5

A-5.6.4 Statecharts

Diagram 48 Statechart for the semi-trailer in case 5
A-5.7 Case 6

A-5.7.1 Class diagrams

![Diagram 49 Class diagram for Case 6]
A-5.7.2 Use case diagrams

**Arrival of goods**

- Production personnel
- Haulier
- Forklift driver
- Card file operator
- High lifter forklift driver

Diagram 50 Use case diagram of the arrival of pallets in Case 6

**Fetch products**

- High lifter forklift driver
- Forklift driver
- Card file operator

Diagram 51 Use case diagram for the fetching of products in Case 6
A-5.7.3 Sequence diagrams

Diagram 52 Sequence diagram for a customer order of a full pallet in case 6

Diagram 53 Sequence diagram for a customer order of a mixed (consolidated) pallet in case 6
A-5.7.4 Statecharts

Diagram 54 Statechart for the Pallet object as it is released from production and put into storage in case 6.

Diagram 55 Statechart for the internal waybill which is created when the pallet is prepared for the warehouse in case 6. The internal waybill is destroyed once the products are retrieved from the warehouse.
A-5.8 Case 7

A-5.8.1 Class diagrams

Diagram 56 Principal class diagram for the cage in case 7
Diagram 57 Use case diagram for the unloading of a trailer in case 7
Diagram 58 Sequence diagram for the arrival and sorting of computers in case 7
A-5.8.4 Statecharts

Diagram 59 Statechart for the unloading and sorting process in case 7

When unloading of semi-trailer is finished, any incomplete consignments left are put in temporary storage.
A-5.9 Case 8

A-5.9.1 Class diagrams

Diagram 60 Class diagram of Case 8
Diagram 61 Class diagram of Non-physical objects in Case 8
Diagram 62 Class diagram of physical objects in Case 8
Diagram 63 Class diagram of the different actors in Case 8

Diagram 64 Class diagram, receiving side in Case 8
Diagram 65 Class diagram, sending side in Case 8
Diagram 66 Physical classes in Case 8
A-5.9.2 Use case diagrams

Diagram 67 Use case for Consignor company in Case 8

Diagram 68 Use case diagram for the Forwarder company on the sending side in Case 8
Diagram 69 Use case diagram for the Forwarder company on the receiving side in Case 8

Diagram 70 Use case diagram for the Consignee company in Case 8
A-5.9.3 Sequence diagrams

Diagram 71 Sequence diagram for the use case Order products in Case 8

Diagram 72 Sequence diagram for shipping products in Case 8
Diagram 73 Sequence diagram for receiving consignments at collecting terminal in Case 8

Diagram 74 Sequence diagram for receiving consignments at dispersing terminal in Case 8
Diagram 75  Sequence diagram for receiving consignments at consignee in Case 8

A-5.9.4 Statecharts

Diagram 76 Statechart for the transport to destination port in Case 8
Diagram 77 Statechart for the loading of goods onto consignment in Case 8

Diagram 78 Statechart for reception at dispersing terminal in Case 8
Diagram 79 Statechart for reception of goods at consignee in Case 8
APPENDIX 6 – SPECIFICS OF CALCULATION MODEL

This Appendix contains various data that are used in the calculation in Chapter 6.4.

This section contains the data and assumptions that are used to construct the matrix over handling time (Table 30) and switching time (Table 31) that are presented on page 155.

A-6.1.1 Handling time

The data for the handling time have been collected from case 1. The following table is used in the calculation:

<table>
<thead>
<tr>
<th>Goods type</th>
<th>Unloading</th>
<th>Coordination</th>
<th>Handling</th>
<th>Loading</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg +/-</td>
<td>Avg +/-</td>
<td>Avg +/-</td>
<td>Avg +/-</td>
</tr>
<tr>
<td>A  Pallet</td>
<td>1.0 0.3</td>
<td>0.7 0.5</td>
<td>1.5 0.4</td>
<td>1.3 0.2</td>
</tr>
<tr>
<td>B  Pallet w/ DG</td>
<td>1.2 0.3</td>
<td>0.9 0.5</td>
<td>1.7 0.4</td>
<td>1.8 0.2</td>
</tr>
<tr>
<td>C  Pallet, unsorted</td>
<td>1.0 0.3</td>
<td>2.0 0.5</td>
<td>1.5 0.4</td>
<td>1.3 0.2</td>
</tr>
<tr>
<td>D  Not on pallet</td>
<td>1.5 0.5</td>
<td>0.7 0.5</td>
<td>5 2</td>
<td>2 1</td>
</tr>
<tr>
<td>E  Pallet w/ monitoring</td>
<td>1.1 0.3</td>
<td>1.0 0.5</td>
<td>2.5 0.6</td>
<td>1.8 0.4</td>
</tr>
</tbody>
</table>

Goods type A – Pallet:

- Unloading
  - According to the terminal facility operator, there are 45 consignments arriving per hour and gate.
  - 1 pallet per consignment gives 1.33 minutes per pallet. Assuming that a consignment is 1.3 pallets gives 1 minute unloading time per pallet.

- Coordination
  - There are four coordinators working in parallel. They coordinate 1900 consignments per working day (300 minutes from 14.00 to 19.00), which gives 0.63 minutes per consignment.

- Handling
  - The internal handling is performed by 7 terminal operators. They handle 1900 consignments together during a working day (375 minutes from 14.00 to 20.15), which gives 1.38 minutes per consignment.

- Loading
  - Average speed when loading is 35 consignments per hour and gate. 1 pallet per consignment gives 1.71 minutes per pallet. Assuming that a consignment is 1.3 pallets gives 1.3 minutes loading time per pallet.
Goods type B – Pallet with Dangerous goods

- Unloading
  - Because dangerous goods need to be put in a separate area, some extra time is assumed (+0.2 minutes per pallet)

- Coordination
  - Dangerous goods documentation needs to be verified, which is assumed to add +0.2 minutes to the coordination time.

- Handling
  - Dangerous goods is assumed to require +0.2 minutes extra handling time due to special placement.

- Loading
  - Dangerous goods are loaded last, in order to be able to facilitate an audit from the authorities. Therefore, an increase of 0.5 minutes per pallet is assumed.

Goods type C – Pallet, unsorted

- Unloading
  - The same as goods type A

- Coordination
  - This goods type needs to be sorted as well as assigned a destination gate. Therefore, it is assumed that the coordination takes 2 minutes per pallet.

- Handling
  - The same as goods type A

- Loading
  - The same as goods type A

Goods type D – Not on pallet

- Unloading
  - Goods are unloaded manually, this is assumed to take 1.5 minutes for a consignment

- Coordination
  - Once on a pallet, there is no difference to goods type A.

- Handling
  - Sorting and handling take longer. Assumes 5 minutes per consignment.

- Loading
  - Loading takes longer. Assumes 2 minutes.
Goods type E – Pallet with monitoring

- Unloading
  - Because of the monitoring demand, some extra time is assumed (+0.1 minutes per pallet)

- Coordination
  - Because of the monitoring demand, some extra time is assumed (+0.1 minutes per pallet)

- Handling
  - Because of the monitoring demand, some extra time is assumed (+0.1 minutes per pallet)

- Loading
  - Because of the monitoring demand, some extra time is assumed (+0.1 minutes per pallet)

A-6.1.2 Switching time

When switching from one goods type to another, some time may be needed to fetch equipment, make decisions etc. The table below shows the assumptions made for the calculation example:

Table 38 Switching from one goods type to another takes a certain amount of time.

<table>
<thead>
<tr>
<th>To\from goods type</th>
<th>From A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

When switching between goods types that are on pallets, and that do not have monitoring demands (goods types A-C) it is assumed that the switch takes 30 seconds. During this interval it is assumed that a terminal operator identifies the new goods type and processes any special information connected to this goods type before proceeding.

It is assumed that the switch from goods type D (“not on pallet”) to the others requires 2 minutes since equipment must be found and fetched (forklift). The switch to goods type D is assumed to take 1 minute, allowing an additional 30 seconds to put away the equipment.

It is assumed that goods type E (“on pallet with demands for monitoring”) takes longer to switch from and to. When switching from goods type E, 30 seconds are added to allow for the operator to report/make a note of the status of the consignment. When switching to goods
type E, the operator may need to obtain a document or some other reference to the individual consignment to be able to not only monitor but also to report the results.