SMART CONTAINER CONSOLIDATION

bundling hinterland container shipping in a competitive multi-actor setting

Master Thesis
H.R.P.C van Hövell tot Westerflier
Student number: 1118978
hovell@initi8.nl
Systems Engineering, Policy Analysis and Management – Information Architectures
Rotterdam, September 24, 2009
Version 42

Graduation Commission
Professor: prof. dr. ir. A. Verbraeck
First Supervisor: dr. ir. H.J. Honig
Second Supervisor: dr. ir. J. van den Berg
External Supervisor: ir. B.J.A. van de Rakt

SUMMARY

Barge operators transporting containers between Dutch hinterland terminals and terminals in the ports of Antwerp and Rotterdam have the room and the ambition to grow. However, this growth is hard to achieve due to the current economic crisis, congestion in the sea ports and competition from other barge operators and other modes of transportation. Barge operators can create a WIN-WIN situation by increasing cooperation among them: transportation costs can be reduced and the CO_2 emissions required for the transportation of containers can be reduced. This can be done using each other's (excess) capacity; barge operators would bundle their separate container flows. However, such efforts are inter organizational challenges and a potential bundling solution has to recognize the intricacies of such a challenge.

This research investigates such bundling solution for the largest barge operator in the Netherlands: Binnenlandse Container Terminals Nederland (BCTN). The main research question is: *What bundling stereotypes and algorithms can be used to improve the efficiency of hinterland container shipping by means of bundling?* In order to answer this question a number of steps have been taken. The current system has been described; this identified actors, objects, interactions and performance indicators important to the system. The performance of the current system has been evaluated using data analysis and simulation; this provided more insights on the current processes and yard stick for evaluating the effectiveness of bundling. The next step was creating a design for a proof of concept bundling implementation, with which the effectiveness of bundling can be evaluated. The input for the design were requirements uncovered during previous steps, personal communications with BCTN and literature research. The results was a bundling proof of concept. The last step evaluated bundled plans created by the proof of concept using simulation and data analysis.

The bundling proof of concept consists of a number of elements: barge operators, container flows, barges, a bundling network and a planning algorithm. The barge operators involved are the operators internal to BCTN: BCT, CTN, CTT and WIT. Bundling is applied to all flows of these operators between their inland locations and the ports of Antwerp, Rotterdam and Amsterdam, and the barges used to transport them. A line bundling network has been selected as the bundling stereotype. Line bundling means that barge will be making different stops at different locations in the hinterland; no transshipment takes place. Line bundling has been selected because of the low impact and the limited transshipment possibilities. A flexible algorithm was needed because of the unique constraints set by hinterland container shipping. Among many competitive algorithms the Adaptive Large Neighborhood Search algorithm was the most flexible.

By bundling their container flows the costs for hinterland container shipping stabilize. Only in extreme cases will the costs for unbundled and bundled plans converge; in such cases the approaches are interchangeable. Bundled planning is more cost effective than unbundled planning in 95% of all cases. However, bundling comes at a price. The robustness (i.e. reliability) of the plans created using bundled operator planning is much lower, more containers are delivered later; this is bad for the business. Resource allocation becomes much more opportunistic. Opportunistic barge allocation can damage the relation between the barge owner and barge operator. Cost analysis showed that not all barge operators gain directly from bundling, some have more costs than in the unbundled situation. These operators need to be compensated. The last two drawbacks are especially troublesome in a situation in which multiple autonomous barge operators would try to bundle their flows. Bundling improves the efficiency of hinterland container shipping but has issues which need to be addressed before implementation is feasible.

It is recommended that BCTN tries adapts it planning constraints in order to create better plans. The Modality instances of BCTN should be integrated as this would greatly improve the ease of planning. Bundling has added value to BCTN, and an implementation based on the proposed system is feasible – only the robustness of the system needs to be improved. The bundling tool has been designed in such a way that it is flexible enough to be used for different transportation problems. The research yielded that more research is needed in the following fields: cooperation between autonomous barge operators and planning uncertainty and risks.

PREFACE

The research presented in this thesis is the result of months of hard work which by writing these last few words are coming to an end. These months have been extremely educational both personally and professionally. During this research I have experienced moments of intense frustration, utter surprise, but I mostly enjoyed working on it. Conducting this research has been a massive undertaking and could not have been completed without the help of a number of people which I'd like to thank.

First I would like to thank my supervisors for their support and patience. Bas thank you for your direct feedback and your ability to clarify things in seconds. Job thank you for reading & correcting everything, your enthusiasm and for the enjoyable discussions we had on and off topic. Jan, your useful advises and unbiased perspective were invaluable. Alexander, thank you for giving me the opportunity to do my thesis at the Systems Engineering section, your advises and your enthusiasm.

This research could not have been possible without the support of Ian Miller and Marco Melis of INITI8. I would also like to thank my roommate, Michael van Eggermond, for the endless discussions on the subject and for proof reading large parts of this report. Furthermore I would like to thank my colleagues at INITI8, my friends and my family. Thank you all!

Rotterdam September 23th 2009
Herman van Hövell tot Westerflier

- Page left blank intentionally -

TABLE OF CONTENTS

1 INTRODUCTION

This chapter introduces the research presented in this thesis. Section one describes the background of the research. The research problem is formulated in section two. The research goal, questions and demarcation are presented in section three. The fourth section describes the research methodology used for answering the research questions. The fifth, and last, section gives an overview of the report structure and relates this to the research questions and the used methodology.

1.1 BACKGROUND

The Port of Rotterdam (POR) is the largest container transshipment hub in Europe. In 2007 a throughput of more than 10 million Twenty feet Equivalent Units (TEU) was achieved (RPA 2008). Most of the containers handled in the POR have a destination in its hinterland, and need further transportation. Due to its unique location, in the delta of the rivers Rhine and Meuse, the POR has access to very well developed waterways. This makes container shipping a very attractive mode of hinterland transportation.

The use of hinterland shipping seems to be a very effective way of transporting large numbers of containers without burdening the already stressed road infrastructure and the environment. However the current handling of hinterland shipping proves to be extremely difficult. As a result, waiting times at terminals are high and terminal capacity is often underused. This is mainly due to the fact that all parties involved in container handling are autonomous and commercial parties. These parties all try to optimize their own business processes. In practice the Barge Operators (BO) have to request capacity for their barges from each port-side terminal. Terminal Operators (TO) then try to make an optimal (from their own perspective) planning for the treatment of barges. Because of commercial interests, TO's do not communicate their plans with each other. As a result the BO usually ends up with a infeasible planning, in which double bookings, bookings with physically unworkable sailing times or bookings with long waiting times are common. This leads to strategic planning requests coming from barge operators (trying to get the right slot) (Moonen, van de Rakt et al. 2005; Douma, Schuur et al. 2007), and more importantly leads to delays in container handling everywhere in the POR. The general effect of these problems is that the modal split of inland container shipping is actually developing negatively in favor of road transportation. This is a bleak reality considering the current environmental and congestion problems in the Rotterdam area (Port of Rotterdam Authority 2008; Transumo 2009).

Many solutions have been and are being proposed (Moonen, van de Rakt et al. 2005; Horst and Langen 2008; Douma 2009). The key problem with most solutions is that these are being enforced by a single actor or create transparency which is not acceptable for the autonomous entities involved in container handling. Container handling is an inter-organizational challenge and a solution must recognize this. However there are (partial) solutions which avoid these pitfalls. INITI8 has developed the Meer Terminal Afstemming Tool (MTA Tool, also known as SYNCHRON8). The INITI8 MTA Tool is a platform which enables the coordination between TO's and BO's without creating transparency. In a workshop setting a predecessor (APPROACH) has proved to be an acceptable solution, which recognizes the intricacies of planning of container barges in the POR (Moonen, van de Rakt et al. 2005).

The Rotterdam and Antwerp hinterland can be divided into three distinct markets which are served by dedicated BO's. These are the Rhine, Antwerp and the Dutch hinterland; each of these markets accounted for about a third of the total hinterland container shipping in 2003 (CBRB 2003). The Vereniging Inland Terminal Operators Nederland (VITO) is the organization in which most of the Dutch hinterland container terminals are represented. These TO's have dedicated barges which transport containers between their own terminal and the POR, in this sense they are also BO's.

All VITO terminals have the ambition and the room to grow. This growth is hard to achieve, because of the congestion in the POR and because of plans by the Port of Rotterdam Authority (PRA) to create competing extended gates (which are hinterland terminals) outside of the POR area. However by increasing operational cooperation between VITO terminals, VITO can create a WIN-WIN situation in which it contributes by reducing the congestion problems in the POR, in which it can decrease transportation costs and in which it can reduce the emissions of CO_2 required for transporting a container. This can be done by using each other's (excess) transportation capacity. This means that TO's would be bundling their separate container flows and are consolidating these flows. However, such a bundling effort faces the same challenges as a solution to the congestion problems in the port itself.

INITI8 is a consultancy firm that is mainly active in the POR. From 2001 onwards they have been involved in the APPROACH project which was about effective planning and synchronization of barge calls. This has lead to the development of the MTA tool. More recent efforts are the monitoring of VITO

barges calling at Europe Combined Terminal's (ECT) Hartelhaven terminal. INITI8 has been asked by VITO to perform research into the feasibility of bundling for VITO members.

Binnenlandse Container Terminals Nederland (BCTN) is the largest inland terminal operator (ITO) within VITO. BCTN manages four terminals in the Dutch hinterland: Container Terminal Twente (CTT), Container Terminal Nijmegen (CTN), Wansum Intermodal Terminal (WIT) and the Bossche Container Terminal (BCT). BCTN will be the focus the presented research.

1.2 RESEARCH PROBLEM

The section describes the research problem. First an overview of the hinterland container shipping system (HCSS) is given (See Figure 1). This is needed to get a clear picture of the processes involved. The second section will describe the idea of bundling and state the main challenges involved in bundling.

1.2.1 HINTERLAND CONTAINER SHIPPING SYSTEM

Due to the well developed Dutch waterway network, transportation over water is an attractive modality for hauling large volumes of cargo. This was slowly recognized by the container transport market and the Dutch hinterland started to develop from the 90' onwards. In 2003 there were over 20 terminals where more than 1.000.000 TEU was handled (CBRB 2003). Most of these terminals are a member of VITO.

Inland terminals manage their own barges. These barges are either owned or hired by the ITO. In order to provide cheap and regular

Figure 1 Geographical overview of the Container Hinterland Shipping System

transport to and from the ports of Rotterdam and Antwerp, ITO's try to maintain a sailing schedule. These schedules are often disrupted by poor performance of container handling in the POR.

The barges that transport the containers between the terminals execute an ongoing process within the Dutch hinterland waterway network. This network can be modeled like a graph: it contains links which connect nodes (Newman 2003). The links are the waterways which provide physically uninterrupted connections between two nodes. A node is an intersection between multiple links or an endpoint of the network. A terminal is such an endpoint (node). The ongoing transportation process is depicted in figure 2 and consists of two basis sub processes:

1. Handling at a container terminal. The barge arrives at the terminal and has to wait until the resources are available to start handling. Once handling starts the containers which have the terminal as its destination are unloaded and the containers which have another terminal as its destination are loaded on the barge. Seaside and inland terminals are similar, they both have quays, stacks, cranes and personnel. Besides their different locations they have different key properties, such as size of the terminal and volume handled by the terminal.

2. Navigating a waterway link in the waterway network. The overall waterway network are all traversable waterway links and nodes. The waterway network (and its properties) traversable by a barge is a function of the overall waterway network and the dimensions of the barge (the dimensions of the barge limit the links which can be navigated) and possibly other attributes of the barge or its load. The traversable network is also influenced by modifications to the overall networks, blockages are a prime example of this.
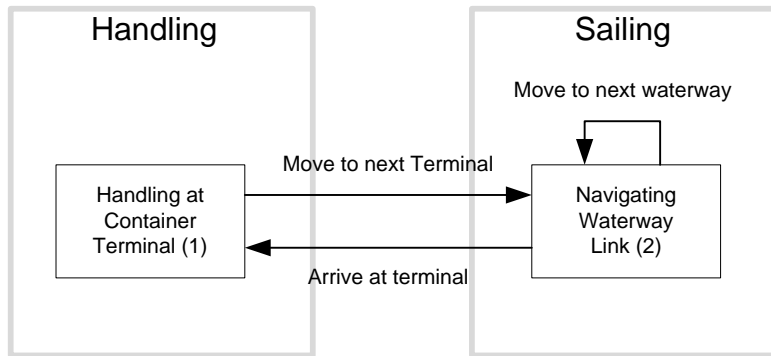
Figure 2 The transport process by barge

1.2.2   BUNDLING

In logistics bundling is a rather broad concept, Konings (2006) defines bundling as:

Bundling can be defined as the common transport of freight belonging to different transport relations in common transport units (e.g. barge vessels) and/or loading units (e.g. containers) during common parts of routes. Basically the intention of bundling is to improve the cost-quality level of transport services, but the characteristics of transport flows (volume and spatial pattern) set the pre-conditions, and there will be a trade-off between advantages and disadvantages.

Bundling is a specific type of consolidation. Consolidation, according to Binsbergen en Visser (2001) is the combination[1] of transport flows in space and time. Consolidation can be achieved by consolidating in time, in route, in place, terminals and in activity. The purpose of consolidation is to increase the utilization of system as a whole and to decrease the use of resources. Bundling is route consolidation focused at the transportation resources, but all other consolidation elements can (to some extend) be found in the definition of bundling. The definition of bundling used by Konings will be used throughout this research. It must be noted that the ideas of consolidation and bundling are as old as transportation itself and their applications remain to be one of the core abilities required for a transport operator (Kreutzberger 2008).

When we apply bundling to the VITO, bundling means that containers from different ITO's which need to be transported from and to VITO terminals will be combined on the barges the VITO members currently hire or own. This will only be attractive when the performance of the ITO's involved improve. The feasibility and the result of bundling is heavily dependent on the goals of the ITO's involved and the constraints set by the system. Bundling for VITO members should have the following advantages:

1. ITO's can improve the performance of their barges. When more cargo can be bundled on the same resource, the barge can become more cost effective. This depends on factors such as sailing distance, sailing time and treatment time of a barge. These factors can change due to bundling.

2. By bundling the containers for a limited set of seaside terminals on a single barge, ITO's can increase call sizes and decrease the number of calls made in the seaside ports. Seaside terminals are the critical resources in the barge's rotation, the waiting time at these terminals is often high. When the waiting time at these terminals remains the same, the time the barge spends in port is reduced.

Bundling does not change the overall complexity of the process. It changes the context of the process, by changing from a single to a multi actor planning process. The ITO will become much more central in the planning process, this means that complexity shifts towards the ITO. Bundling adds and changes the technical and actor specific constraints and requirements within the process. In order to achieve effective bundling these requirements and constraints need to be recognized and taken into account. The main changes to the planning process are described in the two paragraphs below:

The ITO's involved in bundling are autonomous commercial entities. The operators are competitors (on a limited basis), since every terminal services a certain area and these areas have some degree of overlap (Welters, Langen et al. 2003). An important requirement for bundling is the sharing of information on containers (origin, destination, closing and opening times) and barges (available capacity

---

[1] Their definition actually uses the term bundling here, this is omitted in order to prevent confusion.

and availability on routes). This is business sensitive information. The sharing of such information can lead to opportunistic and strategic behavior by the ITO's bundling. A bundling solution should account for this, and must protect the core values of the ITO's involved by reducing the propagation of such information and by reducing the advantages of strategic behavior. From the perspective of the ITO a bundling solution must be trustworthy and create evenly distributed added value. The commitment to bundling and the operational agreements should be institutionalized in order to ensure smooth operation and adequate conflict resolution.

Effective bundling involves the collective scheduling of resources operated by the cooperating ITO's. This is a challenge because the information required for bundling is distributed and has to be exchanged in order to enable bundling and (as became clear in the previous section) because the information should only distributed to the right party. Another remaining challenge is the complexity of the scheduling process itself. The scheduling of N containers on K barges has a huge number of potential solutions. A method should be devised which can resolve a robust (not necessarily the best) solution in a limited amount of time

1.3   RESEARCH QUESTIONS

This section describes the goal of this research and the subsequent research question which should answered in order to achieve the stated goal. The research questions are as precise and concrete as possible. For each (sub-)question a short description is given, stating the goal and anticipated answer. The last part of this section will go into the demarcation of the hinterland shipping system.

1.3.1   GOAL

The previous section argued that bundling, although many challenges are present, can add value for the ITO's cooperating, and that there are even positive effects for other parties such as the Port of Rotterdam Authority (PRA). *The goal of this research is to create a proof of concept bundling tool (from here on PoC) which shows that bundling of containers on shared barges by VITO ITO's is possible, and that doing so increases the efficiency of transportation (decreases transportation costs) without harming the core values of the parties involved.*

While the efficiency of transportation is increased, the research will prove if bundling can be effectively applied to hinterland container shipping. The increase in efficiency can be measured in cost reductions, the effectiveness of bundling can be seen as the size of these cost reduction. However the size of the cost reductions (i.e. the effectiveness) is also dependent on other factors such as the willingness to cooperate between parties.

BCTN will be directly involved as a key member of VITO. Both VITO and INITI8 perceive bundling as an opportunity which needs to be developed. *The goal of INITI8 will be the development of a bundling tool and to gain experience with the design and implementation of bundling tools in general. VITO's goal is get insight in the feasibility of the application of bundling to their processes. These goals are unified in this research.*

1.3.2   QUESTIONS

The research goal is clear, the research question will focus on the design of the PoC, this leads to the following research question:

What bundling stereotypes and algorithms can be used to improve the efficiency of hinterland container shipping by means of bundling?

This question should be answered with the design of a bundling tool and the proof that it improves the efficiency of hinterland container shipping. This question cannot be answered in a single statement and has many aspects to it. The main question consists of the following sub-questions, each of these questions goes into an aspect of the main question, and the synthesis of the answered sub-questions will answer the main research question:

1.   *What are the costs of hinterland container shipping using the current, single operator, planning method?*

The goal of the research is to improve on the current situation by applying bundling to the HCSS. In order to measure an improvement the costs of the current planning method need be known. The execution of a plan produced by the current planning method results in costs: costs for transportation, costs for the environment, etc. The costs might be different for multiple actors. The answer of this research question is an overview of the costs using the current planning method.

2. Which bundling stereotypes are suitable for container hinterland shipping?

There are multiple ways (stereotypes) of organizing the transport network for bundling cargo. The suitability of a stereotype depends on physical, network and actor constraints. This question is aimed to select stereotypes for application in hinterland container shipping. The implementation of a stereotype determines which transportation components are used and how they are used to achieve bundling. The result will the selection of one of these stereotypes for implementation.

3. What bundling algorithms can be used in the implementation of a bundling tool?

The difference between a bundling algorithm and an stereotype is that the algorithm implements the stereotype and makes it available for usage. The choice for a bundling algorithm should be based on the previously selected stereotype, requirements and constraints. The answer to this question should be the comparison between these algorithms and the selection of one of these techniques for implementation in a tool.

4. What are the costs of hinterland container shipping using bundled operator planning?

The costs as a result of the bundled operator planning show the effectiveness of bundling in comparison to current, single operator, planning approach. The costs do not only entail costs in terms of money but also the intended and side effects of bundling (such effects can eventually be monetized). The answer to the research question is an overview of the costs (& effects) using a bundled operator planning method.
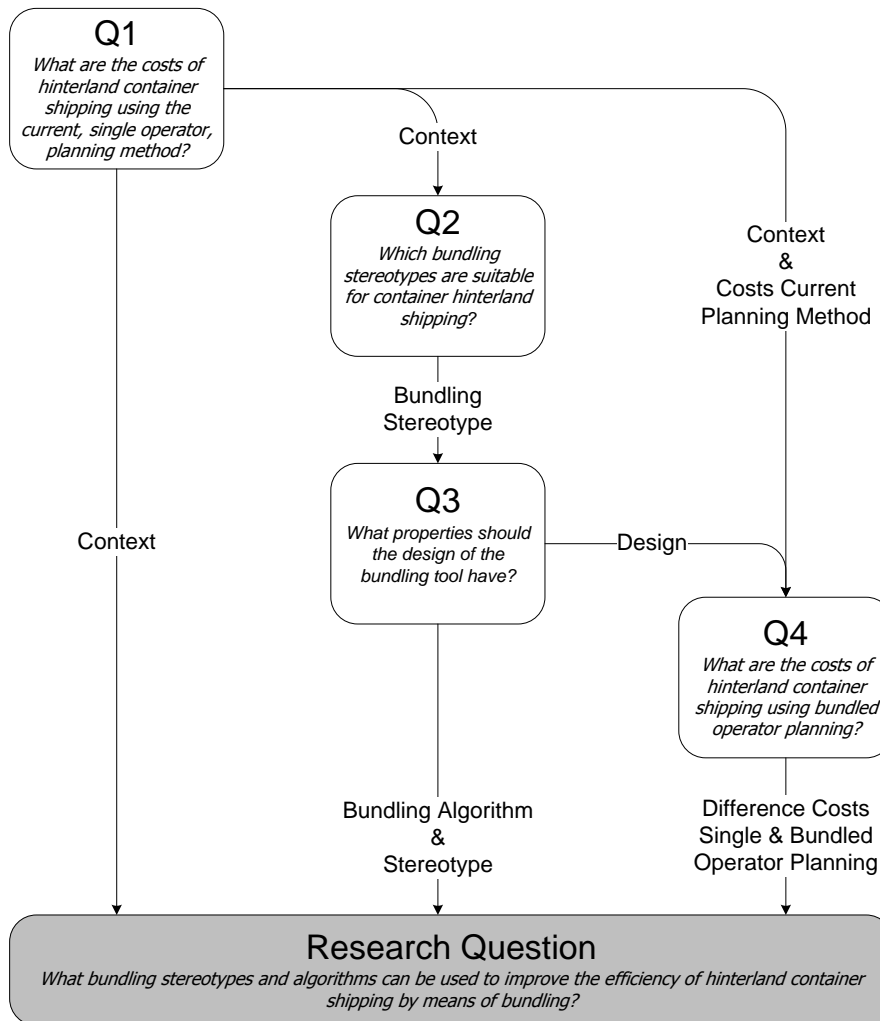

Figure 3 The roles of the sub questions in answering the main question

1.3.3   ANSWERING THE MAIN RESEARCH QUESTION

The answer to the research questions are either a basis for answering a subsequent research question or answer an aspect of the main research question. The final answer, if there is one, is a synthesis of the answer to all research questions. The roles of the sub questions in answering the main question has been depicted in Figure 3, and are elaborated upon below:

1. *Question one* describes the current situation and determines the costs of the current planning method. This in itself is an important contribution in answering the main research question because it defines the system which has been under consideration and provides its context. Question one is the basis for all further research questions.

2. *Question two* finds the bundling stereotypes which have to most merit within the context of the current system. The answer to question two is an important input for the design phase.

3. *Question three* presents the algorithm which can be used to implement a bundling organizational design. The algorithm found answers an aspect of the main research question. The implementation of the algorithm will help to answer the fourth research question.

4. *Question four* determines the costs of hinterland container shipping using a bundling implementation (using a selected bundling stereotype and algorithm). This will evaluate if a bundling implementation will improve the current situation, and answers this aspect of the main research question. To answer this question a yardstick is needed to compare by, this will be provided by the answer to question one.

1.3.4   DEMARCATION

The previous sections have explained the scope of the problem, but haven't clearly defined what will be and what won't be under consideration during the research. The system should only contain the elements which are relevant for answering the research questions, this is done during the demarcation process. This helps to keep the focus of the research clear. It also makes the goal of the research attainable within the given constraints of time and resources. This leads to the following demarcation:
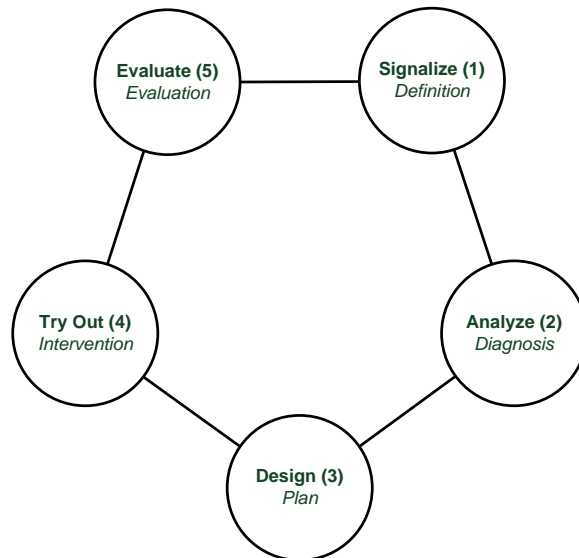
1. The result will be a *proof of concept bundling tool*. It needs to answer the question under which circumstances bundling works. A PoC means that this will not be an implementation ready system. On technical level this means that the core should function, but that other required functionality (e.g. a user interface) will not be implemented. On an institutional level this means that an institutional design is impossible to make, because it is unknown who will be eventually involved. Only a set of institutional principles will be given. Another factor is that the solution space will not be completely explored.

2. It should be noted that bundling can be done on multiple levels: cargo, container and location. For VITO only the container level will be considered. This is due to time constraints and the fact that ITO's have little influence on the customer's process and are neither able nor inclined to stimulate such a bundling effort.

3. VITO is the problem owner, but the actual design will only be tested with BCTN. Other VITO actors will only be involved when a design is actually implemented, in order to validate the design choices made. This is outside the scope of this research. Actors such as the PRA, deep-sea terminals, and the Dutch Ministry of Transport (Ministerie van verkeer en waterstaat - MinVW) will be considered, but only partially because they do have the possibility to influence the outcome but are not central in the research problem.

4. Only a part of the terminal processes will be taken into account during the research. The only part of the terminal process we will consider is the loading and unloading of barges and the availability of quays, cranes and personnel, their reliability, the number of moves per hour and the in- and outflow of containers.

5. Containers will only be partially considered. Weight, size, origin, destination and closing time need to be taken into account. The arrival of containers will have influence on the feasibility of bundling. The arrival process is influenced by economical and geographical factors and cannot be influenced by ITO's. The arrival of containers will be considered a black box and its output will be considered as given fact.

BCTN is the focus of this research. Only barges sailing for BCTN, only the inland terminals operated by BCTN, only the seaside terminals in the ports of Antwerp, Amsterdam and Rotterdam and only the part of the waterway network connecting these terminals will be considered in this research.

1.4 RESEARCH METHODOLOGY

This paragraph will go into the research methods that will be used to answer the research questions. The research will follows a design oriented line of inquiry, the steps of the Regulative Design Cycle (Strien 1986) were used as a guideline for conducting the research:

1. Signalize, the problem is signalized and defined.
2. Analysis, the problem is analyzed and the problem causes are identified and diagnosed.
3. Design, a plan is made.
4. Try out, an intervention based on the plan is made.
5. Evaluation, the intervention is evaluated.

The reason for selecting the regulative design cycle as main research method is the successful application of the method in previous research (Andel 2007; Moonen 2009). It should be noted that the result of the research will be a proof-of-concept bundling tool and not an implemented bundling application. Within the research step 4, try out, is regarded as testing of the PoC in a (simulated) real-world setting and step five, evaluate, is regarded as the evaluation of this test.

Figure 4 The Regulative Design Cycle (Strien 1986)

The regulative design cycle is a high level research method. During the execution of each step lower level research methods will be applied. These research methods will be briefly discussed in the next paragraphs.

Desk research

Desk research will be used extensively during the analysis and design steps. Desk research is also known as secondary research because of the exclusive use of secondary data. This data is gathered from literature, databases, the internet etc. The research will only use the data gathered and reflection on the data to arrive to conclusions (Verschuren and Doorewaard 2007).

Simulation

Simulation is used for the design and try out phases of the research. The simulation process consists of designing a model of the real system and conducting experiments on this model in order to increase understanding of the systems behavior or for the evaluation of strategies for operating the system(Shannon 1975). Simulation will be used to analyze the current situation (step 2), for testing the PoC (step 4) and communication with actors.

The results that simulation will provide have one key property, this is that the model has been constructed from the point of view of the modeler (Sterman 2000), although he or she tries to achieve objectivity through validation and verification it will always have some bias. This must be kept in mind throughout the research, and validation of the findings by experts both from an academic and operational background is vital for the result being meaningful.

Software Prototyping

For the creation of the proof of concept the software prototyping methodology is used (Sommerville 2006 Chapter 17). This is sufficient because we only need a proof of concept and not a fully functional bundling tool. The method starts by uncovering the basic requirements. Then an initial prototype is constructed. This prototype is evaluated. The evaluation will lead to revised requirements and the process restarts. This incremental approach yields a PoC quickly and is used during the design phase.

Supporting methods

Although the above methods are central, these are supported by other methods. The following supporting methods are applied:

1. Literature Research, this will be used to create a theoretical background on the case, bundling methods, simulation and workshops. This is mainly used for desk research.

2. Field Work, for example visiting a container terminal. This will be used to gain more insight in the process of the hinterland container shipping case and to create and validate an ontological model of the case environment;

3. Interviews with experts and involved parties. This will be used to for gaining detailed insight in the case and specifically the actor perspectives, it will also be used for expert validation;

4. Data Analysis of data of the current hinterland container shipping process. This is used to gain insight in the case, but it will also be used as input for the simulation model and bundling tool. The data used comes from INIT8, the POR and BCTN.

1.5   THESIS STRUCTURE

The last section of this introducing chapter presents the structure of this thesis. The structure is aligned with the steps of the regulative design structure, although some exceptions have been made. Each chapter is shortly described and is placed within the context of the research. The overall structure of the research is depicted by Figure 5.

1.   Introduction

The introduction, this chapter, introduces the research problem and underlines the urgency of the suggested research. The introduction completes the signalize step of the regulative design cycle. Furthermore the introduction outlines the research and the report it is part of. This means that the research questions will be presented, the methodology to answer these questions and the demarcation of the problem at hand.

2.   Conceptualization

The conceptualization describes the environment of the system under consideration, its context, and the system itself. The conceptualization is part of the analyze step of the regulative design cycle. The goal of the conceptualization is twofold: on the one hand it serves as a very concise demarcation and on the other hand it provides a qualitative model of the system and a foundation for the research.

3.   Simulating the Hinterland Container Shipping System

The chapter on the hinterland container shipping system model describes the simulation model made for the analysis of the hinterland container shipping system. The model is used during the analysis step of the regulative design cycle, to analyze the current situation. And it is used in the try-out step, to analyze future situations. The goal of this chapter is to provide a concise description of the base model used in both cases.

4.   Analysis of the Current Situation

The analysis of the current situation will give a review of the current situation. The analysis chapter completes the analyze step in the regulative design cycle. This review will be used to compare the performance of a proof-of-concept bundling tool and should be viewed as a yardstick. This chapter will answer the first research question.

5.   Bundling, an Overview

This chapter defines the concept of bundling by first analyzing it from a theoretical perspective, and by then applying it to real-world examples. This will lead to a formal definition of bundling and an overview of the properties of currently used bundling systems. This chapter marks the beginning of the design step, because it defines the design space.

6.   Bundling applied to the Hinterland Container Shipping System

This chapter describes the application of bundling to the hinterland container shipping system. In order to apply bundling an organizational design and an algorithm are selected. The selection of a design and an algorithm answers research questions 2 and 3. The chapter will describe the design of the proof-of-concept bundling tool. It will defend the choices made and will explain in some detail how the tool creates the bundled plans and shows where a proof-of-concept ends and a full implementation begins. This completes the design step in the regulative design cycle.

7. *Evaluation Bundling*

This chapter will give an overview of the effects of bundling on the performance hinterland shipping process. Based on these effects research question four is answered. The evaluation both executes the try-out and evaluation steps of the regulative design cycle. This can be done in one step because the PoC is tested in a controlled environment and not in the real world.

8. *Conclusions & Recommendations*

In this chapter the main research question will be answered. This will be synthesized from the answers to all sub-questions. Based on the insights gained during the research further research is suggested into both the existing research path and new avenues.

9. *Reflection*

The last chapter reflects on the research as a whole and tries to check if the chosen path has been the right one. It tries to see if the research has been answering the right questions, that the method used to answer these questions was appropriate and that the system under consideration was the right system.
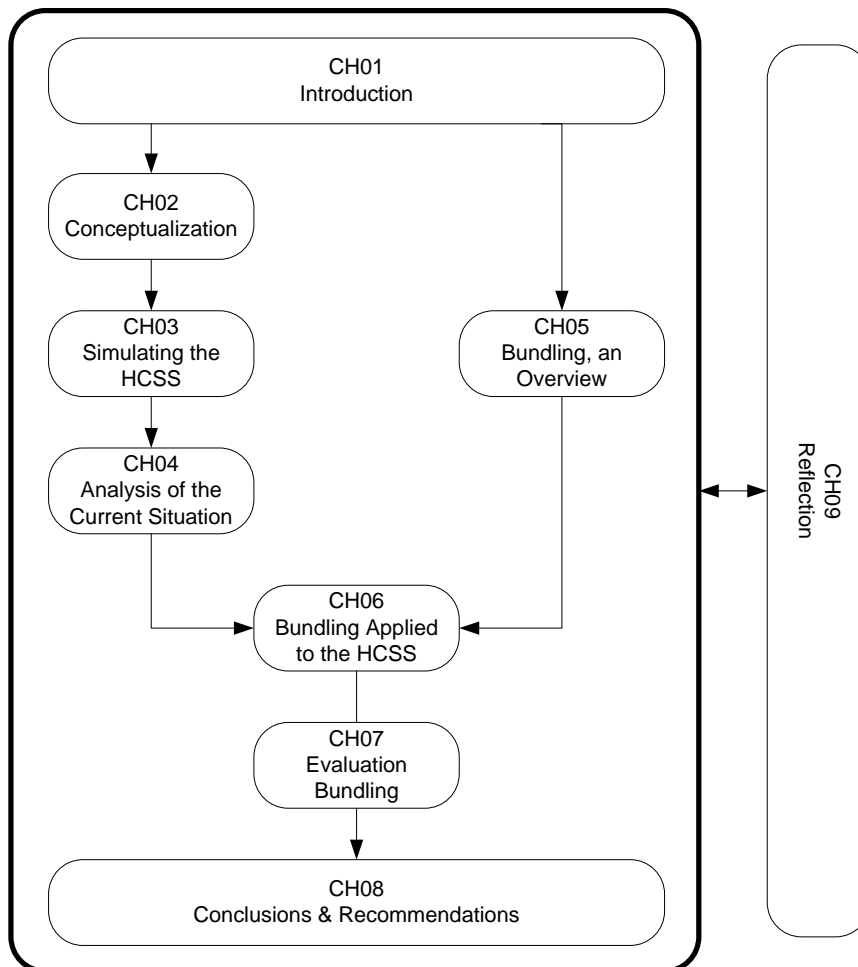
Figure 5 Structure of the Thesis

2 CONCEPTUALIZATION

This chapter describes a conceptual model of the hinterland container shipping system (HCSS). This chapter serves as a solid foundation for answering the research questions and demarcates the research even further by making the system more transparent. This is done by the identifying of the key processes, objects, requirements and constraints of the HCSS. The first section of this chapter deals with the context of the HCSS (the system it is part of). It shows why there is a need for such transportation functionality and it shows where the boundaries of the system are. The second section will delve into the system itself and uncover how the structure of the system creates the required transportation functionality. This section identifies the main actors, objects and their interactions. In the last section the systems' final demarcation will be presented.

This chapter is based on literature reviews, field work and expert interviews. The presentation of the chapter in some areas is supported by the usage of Unified Modeling Language (UML) models (Miller 2003).

2.1 CONTEXT

The HCSS is an artifact of the global supply chain. The global supply chain consists of different steps executed by different actors in which each step adds value to goods by 'transforming' them. The HCSS has been established to fulfill the need for the physical movement of certain types of goods between different steps in the supply chain. The physical moment from origin to destination and all of its subsidiary activities (storage, transfer and data processing) is called transport.

The transport within the supply chain, for example from supplier to customer (as depicted in Figure 6) can be executed by different modes of transport. Based on the type, origin and destination of the cargo, the availability of transport and the quality demands for the transport, different types and modes of transport are possible. Note that de choice for a transport mode can change over time because of changes in quality levels of a transport mode.

Suppliers	*Transport*	Manufacturers	*Transport*	Distribution Centers	*Transport*	Retailer	*Pick-Up*	Customer

Figure 6 A supply-chain from supplier to customer. Transportation links the steps in the supply chain (Adapted from Andel 2007)

The mode used for transport is mainly dependent on the type of cargo, the distance the cargo has to travel, the volume of the cargo, the value of the cargo and the time that is available for transport. There are many combinations of mode and cargo type. Deep-sea transport is almost always used for the transportation of large volumes of similar goods over large (inter-continental) distances. The vessels used for deep-sea shipping are always targeted at a specific type of cargo such as parcel-tankers for liquids, bulk-carriers for bulk cargo and container ships for containers. Air transport is also a long distance transport type but is only used for perishable or high valued goods. For continental transport rail, truck and barges are used for transport. Container shipping is the focus of this research and will be further described in the next sections.

Containerization

Since its initial introduction in the 1950's the container has become the de facto standard for the transport of almost all non bulk goods. The advantage of containerized transport is standardization. A basic container is a metal box with the standard dimensions 20 x 6 x 6 feet which is called a Twenty feet Equivalent Unit (TEU), other forms and sizes are possible but are always derived from this standard. Because the size is standardized, transporting and transshipment resources can be specifically adjusted for container transport which allows for easy transport and transshipment. When cargo is being offered in a container it can always be transported and transshipped by such resources, and multiple parties offering similar services become interchangeable. This allows for competition and dramatically reduced transport costs of non-bulk goods. The reduction in transportation costs made Asian low wage economies (especially China) attractive places to move production to and led to a further containerization of the world. This is one of the major driving forces behind globalization.

The main mode of global container transport is by sea. Although shipping is a relatively slow mode of transportation, it is much more scalable than other modes suitable for container transport. The current trend in deep-sea container shipping is creating further economies of scale by increasing vessel size. These vessels are operated by deep-sea carriers, which organize the schedules of these ships in loops in

order to provide frequent and cost-effective transport. When a vessel is executing such a loop it calls at an ongoing fixed sequence of ports. The ports called by these vessels are usually hubs within the transport network. This means they have very well developed transport connections to other hubs and their less significant spokes. Examples of spokes are hinterland connections. Carriers try to keep the number of stops in loop to a minimum in order to provide cost efficient transportation.
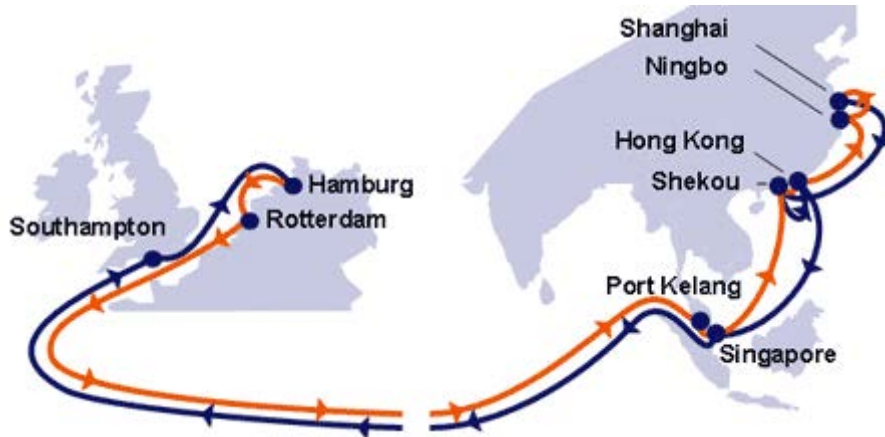
Figure 7 Hapag-Lloyd's EU3 Loop (Hapag-Lloyd 2008)

The Ports of Antwerp, Rotterdam and Amsterdam

The ports of Antwerp, Rotterdam and Amsterdam (ARA-ports) together are among the largest container transshipment hubs in western Europe, their combined throughput in 2007 (Rotterdam Port Authority 2008) was more than 19 million TEU's in 2007 (which is 49.5 %[2] of the total throughput for the Hamburg-Le Havre range of ports). Due to their geographical position the ARA-ports have a shared hinterland. Deep-sea operators use this in their advantage and usually only add one of these ports to their loops. This is a cause for fierce competition between these ports. The vessels calling at these ports will transship their containers at dedicated deep-sea terminals, such as ECT or MSC Home. Containers do not stay at these terminals but are transported to their destinations in the port's hinterland[3]. The destination, the quality of service and the availability of transport determine if trucks, rail of barges are used for further hinterland transportation.

Hinterland Container Transportation

The unique position of the ARA-Ports in the estuary of the Meuse and Rhine rivers and the well developed waterway network makes transport per container barge an attractive option for the hinterland transport of containers. There are three modes of transportation suitable for hinterland transport; these are transport by train, truck or barge. In 2007 the modal split between these modes for the port of Rotterdam was 11% for transport by train, 59% for transport by truck and 30% for transport by barge (Rotterdam Port Authority 2008). The quality of infrastructure and the treatment process for a modality has a large influence on the development of the modal split for a modality. Hamburg (and Germany as a whole) has a better developed rail network, and hinterland transport by train is therefore better developed.

Hinterland Container Shipping

Seeing that 30% of the hinterland container transport for the port of Rotterdam is done by barge, it is useful to understand the major advantages and disadvantages to hinterland container shipping (Port of Rotterdam Authority 2008):

Advantages:

1. Economies of scale. Barges can haul large amounts of cargo in comparison to transport by truck and train. The most used barge type measures about 110 meter by 11 meters and can haul a maximum 208 TEU, while a truck can haul a maximum of 4 TEU and a train with the maximum

---

[2] The ports of Rotterdam and Antwerp are the large deep-sea container ports, Amsterdam accounts for about 1% in 2007.
[3] The text has been written with transport from the deep-sea vessels to the hinterland in mind. There is also a transport flow from the hinterland to deep-sea transport. But for the sake of readability only one of these flows is described.

length of 618 Meters can haul 90 TEU (Roscam Abbing 1999; Centraal Bureau voor de Rijn- en Binnenvaart 2003).

2. Hinterland shipping uses the waterway network and does not add further strain on the road and rail network. The road and rail networks in the Netherlands are among the most used networks in the Netherlands, while the waterways still have ample capacity.

3. The emissions of CO2 are relatively low in comparison to other modalities (Planco 2007).

Disadvantages:

1. Hinterland container shipping mainly requires another form of transport which will deliver the container at the client. This adds complexity to the transport and requires an additional transshipment, this takes time, resources and money.

2. Barges use the same quay and crane capacity at deep-sea terminals as deep-sea vessels. This conflict of interest is usually resolved in favor of the deep-sea vessel, because stevedores have a contractual relation with the carriers and not with the barge operators.

3. Transport by barge is inherently not the fastest mode of transport available. For some JIT deliveries transportation by barge is too slow. This is however a relative statement, it usually takes containers a month to get from origin to destination and a relatively small amount of this time is spent during hinterland transport (Roscam Abbing 1999).

4. Hinterland container shipping requires a well developed waterway network and inland terminals for further transshipment of containers. This is the case in the Netherlands and Germany, so in those countries the disadvantage is void.

The Hinterland Container Shipping Market

The hinterland of the ARA-Ports can be divided into three distinct service areas. Although these service areas use the same mode of transport, different parties are active within these service areas and there is little to no overlap between them. The service areas/markets are described in the table below, it should be noted that the market shares given are from a 2003 CBRB publication(CBRB 2003), and are a mere indication of the market size:

Table 1 Hinterland Container Shipping Service Area's

Service Area	Description	Size
Antwerp-Rotterdam Feeder Connections	The ports of Rotterdam and Antwerp exchange large volumes of containers each year. This is mainly because the loops of carriers usually only have one of these ports in their loops, containers for the other are exchanged by these feeder lines. The transport in this market is characterized by the short travel distance, large call sizes and few calls made.	26%
Rhine Connections	The Rhine provides an excellent natural corridor for transport to and from Germany. There are a large number of terminals situated along the Rhine and its tributaries (Main, Neckar and Mosel). This even makes connections from Rotterdam to Switzerland possible. This service area can furthermore be divided into three sub-areas, the Lower-, Middle and Upper Rhine. The transport can be characterized as long distance with a relative high number of calls.	37%
Inland Terminal Connections	Due to its unique geographical position, waterways in the Low Countries are very well developed and major parts of the Netherlands and some of Belgium can be reached by water. This has lead to the development of container terminals along the waterways. The terminals all have connections with the ARA-Ports. This transport can be characterized as short/medium distance with the most calls made in seaports.	37%

There are about 20 container terminals suitable for container barge handling located in the in the Netherlands. Most of the TO's running these terminals are part of VITO. These terminals are scattered throughout the Netherlands. BCTN as a terminal operator manages four of these terminals.

BCTN

BCTN transports goods between their inland locations and the ports of Antwerp, Rotterdam and Amsterdam (ARA-Ports). In some instances BCTN organizes transports between their terminals and Rhine of other inland terminals, however occasional this extends the operational geographical area of BCTN operations to the Rhine and other hinterland connections.

Table 2 BCTN Managed Terminals

Name		*Location*	*Waterway*	*Transshipped*
Bossche Container Terminal	BCT	Den Bosch	Meuse	105.000 TEU
Container Terminal Nijmegen	CTN	Nijmegen	Waal	75.000 TEU
Wansum Intermodal Terminal	WIT	Wansum	Meuse	70.000 TEU
Container Terminal Twente	CTT	Hengelo	Twente Kanaal	90.000 TEU

BCTN transports containers between their inland locations and the ARA-Ports. Occasionally BCTN organizes transports between their terminals and Rhine or other inland terminals; this extends the operational geographical area of BCTN operations to the Rhine and other hinterland connections.

In order to provide the transport required BCTN manages a fleet of approximately 12 barges. The barges are not owned by BTCN but are hired for long periods of time. The rent paid for these barge is bare, this means that the price for a contractual period is fixed and that BCTN pays the port dues and the fuel costs for these barges (BCTN 2009).

Transportation by barge is never a door-to-door mode, to achieve door-to-door transportation BTCN complements their fleet of 12 barges with a fleet of owned and hired trucks (depends on the terminal).

System Overview

This section zoomed in on the HCSS placing it in the context of the global supply chain which it is a part of. This has created a clear understanding of the role of hinterland container shipping. By zooming in it becomes apparent that the work presented in this thesis borders which can be linked to the key actor within this thesis, BCTN. Only the HCSS for BCTN will be under consideration. For the sake of scalability it must be pointed out that other operators have very similar processes, and that only small parts will vary from operator to operator. Because of this a number of system borders can be defined:

Table 3 System borders from BCTN's point of view

Type	*Demarcation*
Geographical	Only ports, waterways which are regularly (no occasional transport) utilized by BCTN will be considered. This in effect means that only the ARA-ports, the ports at which the inland terminals are situated and their connecting waterways are considered.
In- & Output	The purpose of the container hinterland shipping process is to transport a container from terminal A to terminal B. Only container transported by and thus booked at BCTN will be considered during the research.
Actors	The actors to be considered are only the actors with whom BCTN interacts. More on this in the next section. These are clients, shippers, shipping agents, barge operators, carriers, stevedores, customs, barge captains, inland terminals operators, local transporters and receivers.
Objects	Objects are limited to the objects BCTN interacts with. These are own barges, containers transported by BTCN, waterways used by their barges and terminals

visited.

At this point (BCTN's) hinterland container shipping process is a black box. Its functionality is known, but not how the process is implemented. The following sections are going to analyze the process and make the black box white where possible and necessary. A more content focused demarcation will be provided during these sections.

2.2 HINTERLAND CONTAINER SHIPPING SYSTEM

This section describes the hinterland container shipping system (HCSS) in more detail. The previous chapter described what the functionality of the HCSS is. This section shows the structure of the process. The structure of the process consists of the objects, actors and their interactions. The actors will be described in the first section, the second section describes the actual system and the interactions of the actors with the system.

2.2.1 ACTORS

The actors within the process shape and operate the current process. The objectives, perspectives, interdependencies and means of the actors determine how the process works, what requirements and constraints there are to the process. Figure 8 places the actors within the shipping process. This section presents the results of an actor analysis, which can be found in Appendix A. This section will go into detail on the key actors in hinterland container shipping, others are only involved indirectly (Rakt 2002; Connekt 2003).

Table 4 Key Actors, theirs interests, goal, criteria and powers

Actor	Goals	Criterion	Powers
Inland Terminal Operator	Efficient Container Transport Reliable Container Transport	MIN(Costs/Container) MIN(#Containers missing closing times)	Change planning methods Use different modalities Cooperation Change Capacity
Barge Owner	Keeping to contract regulations	MIN(Deviation from contract criteria)	Negotiating contract terms
Shipper	Efficient transport Satisfying transport quality	MIN(Price/Container) MIN(Lateness)	Change transport mode/parties.
Stevedore	Efficient terminal operation.	MIN(Costs/Move)	Change the demands for treatment Change capacity

The goal of this research is to create a PoC for the HCSS. The bundling of container flows will be primarily executed by Inland Terminal Operators which are also Barge Operators[4]. Barge Owners and the Shippers are especially affected by this since the patterns in which containers flow will change. To a lesser extent or in special cases stevedores will also be affected by bundling. These four actors are central in designing a feasible bundling system and their goals, powers and criteria (These criteria have been based on Goor and Ploos van Amstel 2002) are considered during the research. These are elaborated upon in the Table 4 (above).

---

[4] From here on the term Inland Terminal Operator will refer to the combination of the Inland Terminal Operator Role and the Barge Operator role.
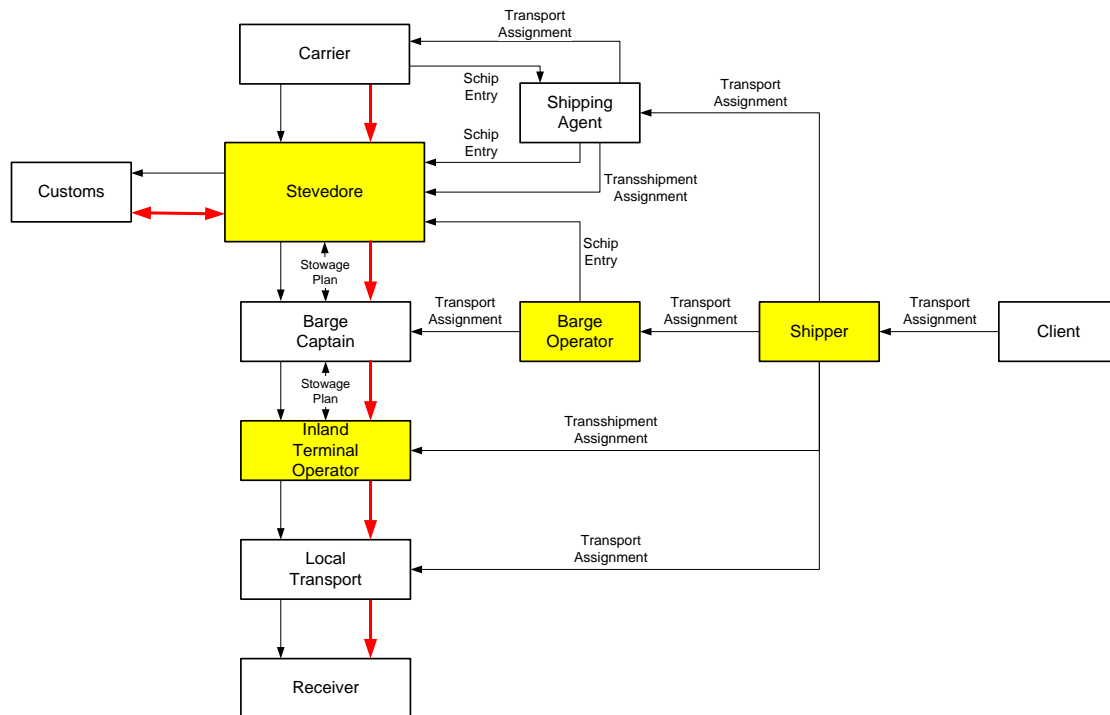
Figure 8 Actors involved in the hinterland container shipping process[5] (Adapted from CBRB 2003)

2.2.2 LAYERS

The system consists of a limited number of interacting objects. However, the system can yield very complex and very unexpected behavior due to the large number of objects and the constraints governing the interactions between objects. The previous section describes which actors influence which objects and interactions. This section describes the actual interactions and objects. In order to do so, the system has been divided into a number of layers, similar to those of for example the Open Systems Interconnection reference model (Binsbergen and Visser 2001; Stallings 2001). These layers are hierarchical and each layer provides functionality for the next layer (Dietz 2006).

The base layer within the HCSS is the physical layer; this is the infrastructure of waterways. The infrastructure is what is used by barges to transport containers between terminals. The use of the barges is called the transportation layer and describes the physical movement of containers through the system. The transport layer and its processes are controlled by the planning layer. This layer take contains the informational and communication processes which orchestrate (plan) container transportation.

Each of these layers is described in a dedicated section below. A textual and a visual description are given for each of these layers. The visual representation is done using general diagramming techniques and the Unified Modeling Language (UML), class diagrams are used to describe the objects in the system and sequence diagrams are used to describe the (main) interactions in the system.

2.2.2.1 INFRASTRUCTURE LAYER

The physical layer contains the infrastructure that enables the transportation of goods (containers) by barge between two terminals. The physical layer consists of a network, the hinterland shipping network (HSN), and vessels interacting with it. The network connects origin terminals to destination terminals by means of interconnecting waterways (links) and nodes. Vessels navigate the network by traversing waterways.

---

[5] The unlabelled thick (red) lines follow the flow of containers, the unlabelled thin (black) lines follow the flow of information. This figure depicts the case in which the client is receiving cargo. The key actors are marked yellow (only the barge owner is not directly involved in the transportation process)

Waterway Link

A waterway link (an undirected edge in the graph that models the transport network) connects two nodes in the network. A link is a geographical entity which is usually part of a river of channel, for example the river Meuse. Waterway links have some capacity restrictions but they usually apply to the dimensions of the vessels traversing them. Within the research bridges (road or rail connections crossing the waterways) are not explicitly modeled. They add another restriction to the maximum passable dimensions of the waterway and influence the travel times of barges. Currently waterways are not susceptible to congestion, so no capacity considerations are made. However link availability can be an issue and will be taken into account, an example of this is the blockage of the Rhine near Cologne in 2007 (Inspectie Verkeer en Waterstaat 2008). Waterway links (especially near sea-ports) have a current, this means that link traversal for different directions yield different results in terms of speed and time, and can be time dependent.

There is one special case of a waterway link, and that is a link which is part of a lock system and is assigned to a lock chamber. Capacity on these links is restricted, and must be requested in advance.

Node

A waterway node (also called a vertex in graph theory) is the connecting point for a number of links. There are a few different node variants, but they all have this connecting property in common:

1. Intersection Node, this node is just merely used to connect two or more links.
2. Quay Node, this node is an endpoint in the network and usually only has one link attached to it. A quay is part of a terminal. From a modeling perspective, the most important property of the quay is that it has a capacity associated with it, which must be requested. This capacity can be partially modified by the terminal operator (the length of the quay will almost always be the same) by scheduling cranes and personnel at the quay.

Vessel

A vessel is the object that navigates the network. The vessel has properties such as its length, width, height and draft which limit the waterway links a vessel can use. Draft and height of vessel are influenced by the vessel's weight, the cargo's weight and form factor. The traversable network for a vessel is therefore a function of its dimensions, its weight, its cargo and the network. The vessels speed is dependent on its engine and possible speed restrictions for some of the waterways.

A vessel usually remains within the bounds of the network. A vessel's activities can typically be divided into two main categories: activity at a node (resting, loading, unloading etc...) and movement through the network from an origin node to a destination node, both activity categories have been depicted in Figure 2. The sequence of the links the vessel traverses is the path of the vessel and the traversal of a link is a stage of such a path. Successive stages in a path must be represented by connecting waterway links. A path starts at its origin node and is finished at its destination node. The traversal process is quite straightforward. A vessel starts by entering the initial stage and it keeps moving from stage to stage until it finishes. The time it takes to finish a stage depends on the availability of the link, the length of the link, the current on the link, the heading of the vessel and the speed of the vessel. As stated before, normal links have nearly unlimited capacity so these links can always be entered if the entering conditions are met, lock links require the allocation of the lock chambers' capacity, which can cause delays during the traversal.

2.2.2.2   TRANSPORTATION LAYER

The transport layer describes the physical transport of containers from origin to destination terminal. Transportation is executed by a barge. A barge is a specific type of vessel which is capable of transporting containers. Transporting containers is different from other cargo types because containers are stacked within the hold of a barge. Containers are only allowed to be stacked to a specific number of containers. This is dependent on the size of the barge and is regulated by law (Inspectie Verkeer en Waterstaat 2008). Containers must be available for discharge at a destination terminal. When the container is positioned under another container, the container above it must be moved in order to free up the container, this is called a shifter move. Shifter moves are overhead for the terminal operator unloading the cargo and a barge operator has to pay for them when the number of 'shifters' is too large. Transporting containers requires the barge to be properly stowed. A more subtle problem occurs when the call order of the barge changes during execution, this means the barge is not stowed optimally anymore and this causes even more shifter moves. The stacking height employed increases the height of the barge, as stated in the previous section this limits the network which can be traversed by the barge. An example of this is CTT. CTT mostly uses barges with a capacity of 208 TEU, however

these barges can only carry 104 TEU because passage height of the locks to Hengelo is such that no more that more two stacked containers can be stacked without exceeding these height limitations (the capacity of 208 TEU is based on four stacked containers).

For a vessel two types of activities were defined: activity at a node and movement through the network. Movement through the network for a barge is the same as for a vessel; however the origin and destination nodes are terminal quays. During the activity at the node the barge is loading and discharging containers at a terminal quay, this is a call. The call requires the capacity of the barge and capacity of the terminal quay at which the rendezvous takes place. The quay must have space available, have a crane ready for treatment, personnel available and the containers to be loaded must be at the quay. Without one of these elements transshipment is impossible, and the barge will have to wait or move to another terminal. This is why terminal operators and barge operators must coordinate their actions in order to achieve smooth operations (more on this in the next section). Before and after the transshipment the barge needs to be moored at the quay and this takes some time as well. During the call the crane moves containers from the quay-side on the barge (loading) and vice-versa (discharging) and keeps doing this until all containers are transshipped. For a more detailed description of the calling process the thesis of M. van Andel (Andel 2007) can be consulted. The processes involving the containers on the terminal are outside of the scope of this research. The complete transportation process has been depicted in Figure 9.
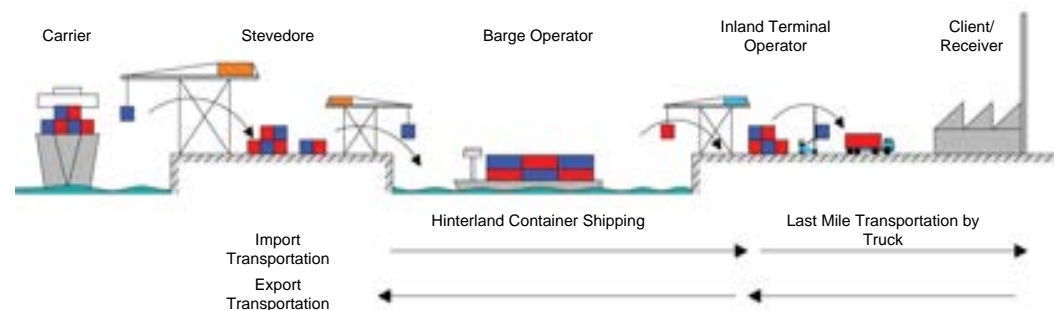
Figure 9 Transportation Process (BCTN 2008)

2.2.2.3   PLANNING LAYER

The planning layer describes the interactions and objects required to plan and execute the transport of containers. Note that there is a significant difference between planning and scheduling from an academic point of view (Smith, Frank et al. 2000). A plan defines which tasks need to be executed. While a schedule defines how these tasks are executed (when and by which resource). It is the latter which is created during the within the 'planning' layer, not the first! The term planning is chosen because it is term most commonly used for such problems within this context.

There are two main processes within the planning layer: the booking process and planning process. These processes take place in a (semi-)concurrent fashion. The layer description follows the container from the initial booking to the finalized planning.
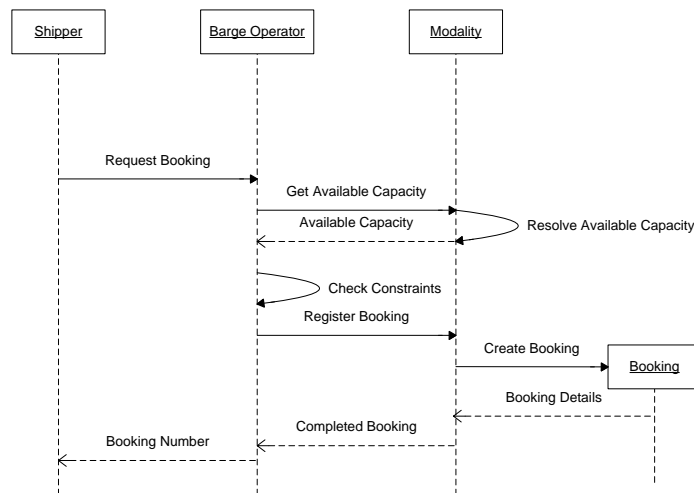
Booking Process

Figure 10 Sequence diagram booking process

The booking process is initiated by the shipper who wants to transport containers using the barge operator's facilities. The shipper requests transport with the operator's booking office. When transport request is feasible a booking is made and confirmed to the shipper. This process is depicted in Figure 10. This booking describes the container transports which need to take place. The booking can also contain information on depot movements, last-leg transportation by track and other things. These are not considered in this research. The booking contains details about the movement of one or more containers from an origin terminal to a destination terminal. The transports also provide details on the availability of the container and the containers cargo and weight, these objects and their relations are shown in Figure 11.
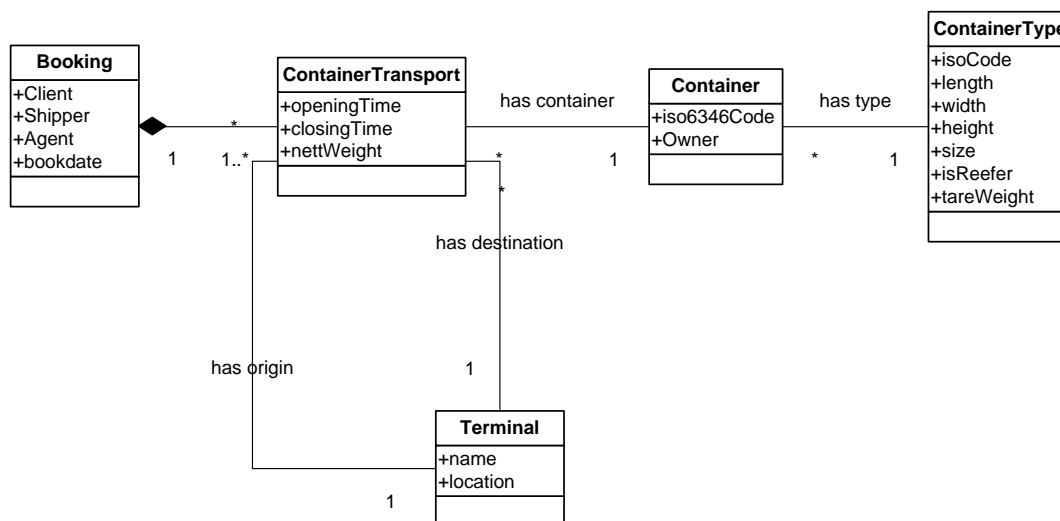
Figure 11 Class diagram bookings

Modality

BCTN, like most barge operators, uses 'Modality' for storing their bookings and as their primary system for conducting operations. Modality is a software package specifically targeted at inland container shipping and allows (among other functionality) an operator to register the bookings and transports, plan barge voyages and send invoices to customers. BCTN currently runs four separate instances of Modality, one for each terminal. Although the planners and bookers can access all instances at the same time, this cannot be done using a single interface. There is no communication between the instances,

so the planner has to switch back and forth between them. In the future the systems of BCT, CTN and WIT will be merged.

Call Planning Process

The booked transports must be planned on the available transportation capacity. This is done by the planners. BTCN currently has four planners: one for each terminal. They work in the same office and can cooperate in order to create synergy in the planning effort. The planning process has been depicted in Figure 12, and consists of these steps:

1. The call planning process starts with container transports which are eligible for transportation. Eligible transports are the container movements for which the containers are available, which can be delivered to their destination and for which the paperwork is in order. When a container has a deep-sea terminal as its destination, the internet is checked by the planners in order to check if the closing and opening times for containers have changed (most deep-sea terminals have websites on which they publish the most recent actual time of arrivals of the deep-sea vessels). Containers cannot be collected or delivered before their opening and after their closing times.
2. A call is created by combining the eligible container transport with the available transportation capacity. These combinations are often found using a number of first-come first-served heuristics. A barge is usually planned for a designated service from a hinterland terminal to a specific number of terminals in a deep-sea port with a given frequency; an example is CTN's Ophir which visits terminals in the Rotterdam Delta area every two days. In this case services are aimed (at parts of) ports, port being a usable aggregate for geographically clustered terminals; a port – to– port is deemed efficient. However, these services are not cast in stone and provide merely a guideline in planning these barges. Calls are planned pragmatically in such a way that they meet the demand for transportation. In some cases the available capacity is unable to satisfy the requested capacity because of the sheer lack of capacity or too strict time constraints. Trucks are used when the capacity demand is low or when the time constraints are extremely tight, and barges are temporarily hired when the demand for capacity is high enough. Transportation by truck and its planning lie outside the scope of this research. This is the only step that needs to change in order to apply bundling to the HCSS.
3. In the next step the call is communicated with the terminal which will execute the call and who has to plan capacity in order to handle the call. The planner sends a capacity request to the terminal planner. This is done using the phone or electronic data interchange (EDI – using EDIFACT or BERMAN messages). During this step the planning processes move from BCTN to the operator being called. This request usually has to be made a certain time before the call is executed (this called a planning horizon). ECT Delta for example uses a planning horizon of 24 hours for planning. This allows the terminal planner to make an effective resource (quay, crane and personnel) planning for the terminal. Calls are planned in such a way that they are optimal for the terminal operator. This means that the start operations time for the call is usually changed in order to optimize the terminal planning. These modified requests are returned to the inland terminal operator (The effects of this planning mechanisms are described in the introduction). Sometimes a call cannot be executed by the terminal and the request is rejected, meaning that the call needs to be re-planned he planners. This can be a big problem for barges whose travel time is larger than the time window presented, these cannot be re-planned.
4. The last step in the coordination process finalizes the planning processes. In this step the BCTN barge planners receive the confirmed (and modified) capacity requests back from the terminal operators. These requests are merged with the planned calls using Modality. Final adjustments are also made during this step. The result of this step is the finalized call, which is sent to the barge captain which will execute the call. Here the planning processes ends and the transportation process commences.
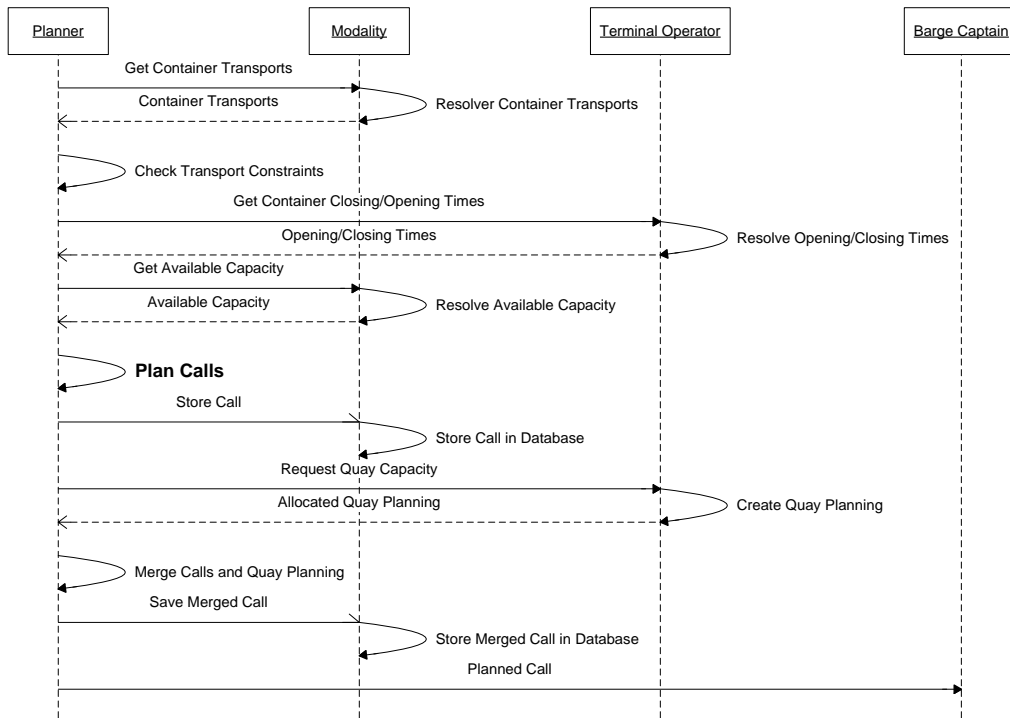
Figure 12 Call planning process

Demarcating the planning layer

Bundling changes the scope of the barge operator's call planning process. The planning processes of multiple barge operators are combined: more container transport and more barges are considered. The planning process will still have the same in- and outputs, and is still subjected to the same constraints. This however does not mean that all processes within the planning layer are relevant within this research.

The outcome of the booking process is an important input for the planning process. It will have a significant influence on the outcome and effectiveness of bundling. The booking process itself will not change, only the resulting container transports will be considered. The two other planning processes, planning by stevedores and barge captains are not changed by the implementation of a different approach to planning. They are subjected to different call patterns and will react accordingly. These processes will be taken into account marginally, because they have an influence on the execution of calls and not on the planning of calls.

2.3   CONCLUSION: THE HINTERLAND CONTAINER SHIPPING SYSTEM

This chapter showed the context in which the HCSS is embedded and the system itself, the results of this chapter are used as a basis for answering the research questions. The focus of this research is changing the planning process within the HCSS and its effects on the system as a whole.

During the system analysis the main planning constraints were uncovered. These constraints should be respected by any planning approach. The following planning constraints have been uncovered:

1. Time constraints, these limit the time in which activities can be conducted:
    a. Container Transport Opening Times, the container to be transported cannot be picked up at its origin until the opening time of the container. This usually depends of the previous (transportation) step taken by container. A container cannot be picked up when it is not ready for transportation or when it is not available, the opening time defines the moment the container becomes available.
    b. Container Transport Closing Times, the container to be transport needs to be delivered at its destination before its closing time. The closing time is dependent of the next step taken by the container. The container must be available at its destination in order to be loaded

onto its next transport or to be treated. The closing time defines the moment the container needs to be delivered.

c. Call Time Windows, this minimum time between the registration of a call and the execution of a call. These are enforced by terminal operators in order to enable them to plan their resources effectively. This means that a call at a terminal with a time window needs to be planned well in advance, because it needs to be communicated with the terminal operator before the start of the time window.

d. Terminal Opening hours, most terminals employ regular working hours. This means that they operate between 7 AM and 7 PM, call planned before or after these opening hours will have to wait.

e. Repositioning time, it takes time to move a barge from one location to another. This means that the travel time of the barge needs to be taken into account while planning calls. The position of barges makes some barges more suitable to fulfill container transports than other barges.

2. Capacity Constraints, these limit the use of capacity within the system:

a. Barge Capacity, the planned containers are not allowed to exceed the barges maximum capacity. Note that the maximum barge capacity can be dictated by the route the barge is taking through the network.

b. Terminal Capacity, a terminal has a limit to the number of barges which can be treated at the same time at its quays. This is dependent on the quay length, the number of cranes available and the planning of crane personnel. When the number of concurrent calls at a terminal exceeds the terminal's capacity, poor treatment time can be expected.

3. Network Constraints these limit which resource can use parts of the network:

a. Size constraints, the size of network objects can limit the size of barges which can pass. The depth and width of waterways limit the barges which can use the waterway. The size of locks can limit the size of barges which can pass. The passable height of bridges and locks can limit the number of containers stacked in a barge.

b. Temporal constraints, these change the use of a network resource. Accidents, maintenance, natural events (drought, tide) can make network object unavailable or severely limit the use of the network resource.

The final perspective on the system is that it can be divided into two main components: a planning component and a transportation (executing) component. These components will be used throughout the research and have been depicted in Figure 13 using an IDEF0 diagram (Defense Acquisition University Press 2001). The planning component transforms transport requests into plans. The transportation component takes these plans and executes them. A plan consists of a number of calls at different terminals by different barges, picking up and delivering containers.
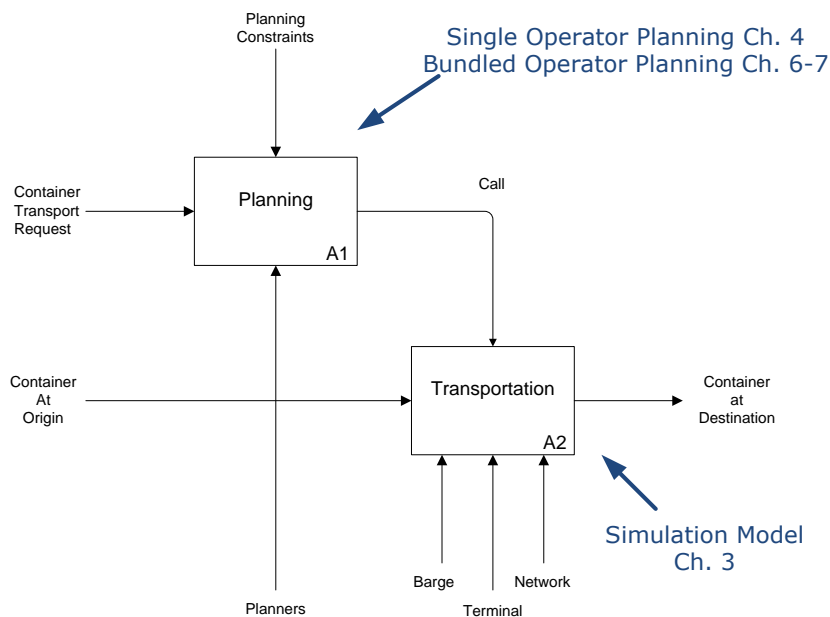
Figure 13 System Perspective

Within the research two types of planning components are used: a single operator planning (SOP) component, which replicates the current planning approach (this is discussed in chapter 4) and a bundled operator planning (BOP) component which can bundle the requests for difference barge operator into combined calls. The implementation of this planning component is discussed in chapter 6 and the results are discussed in chapter 7.

3   SIMULATING THE HINTERLAND CONTAINER SHIPPING SYSTEM

The previous chapter introduced the most important objects, actors and interactions within the HCSS. A simulation model of the HCSS has been constructed in order to be able to analyze the 'as-is' performance and the 'to-be' performance (using bundling) and their differences. This model has been constructed based on the concepts introduced in the previous chapter and has been filled with data gathered from different sources.

This chapter contains a global description of the constructed model. Because the model is used for both the analysis of the current and future situations, the description of the experiments performed using the model can be found in the chapters analyzing the current (chapter 4) and future (chapter 7) situations.
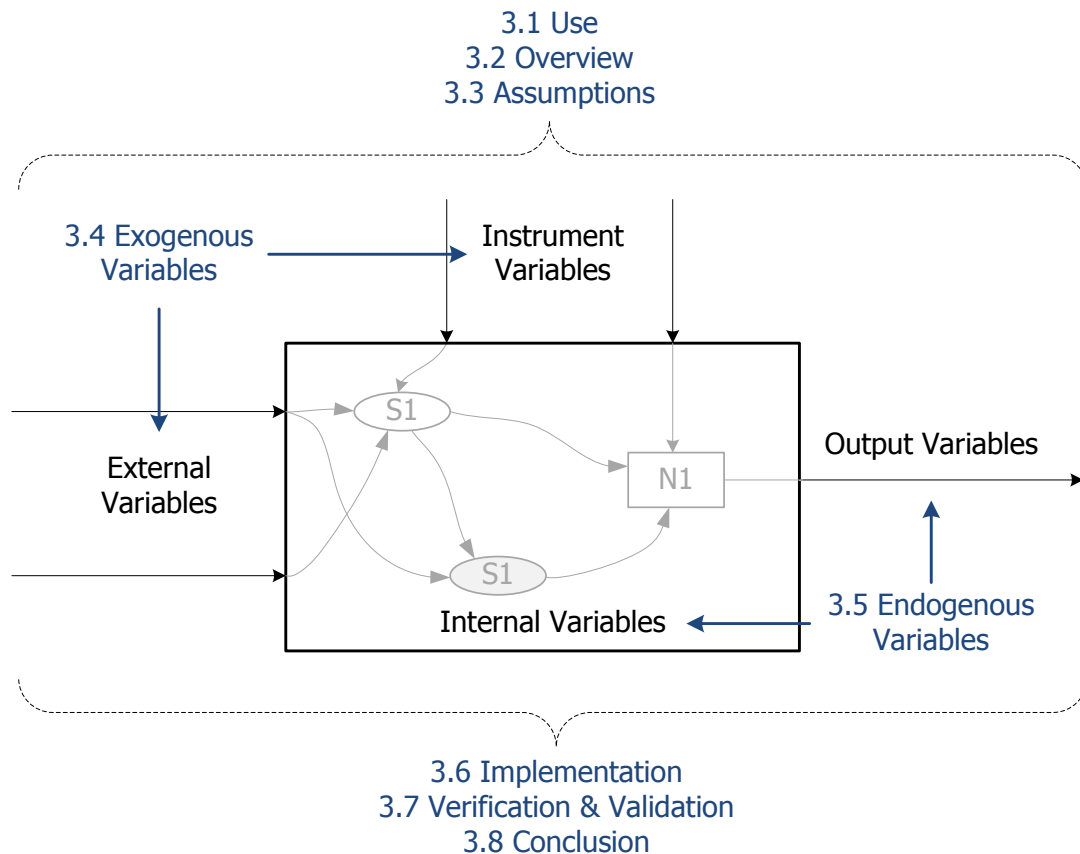
Figure 14 Overview Chapter 3 (adapted from Bots 2002)

The use of the model and its role in the analysis of the system will be discussed in section 3.1. An overview of the model will be given in section 3.2. A model is a simplification of reality, section 3.3 describes these simplifications and the assumptions made in the model. Section 3.4 describes the exogenous variables which are not determined by the model, and in fact drive the model, these are the instrument and external variables. The variables that are modified by the model, the endogenous variables, are described in section 3.5. A short overview of the implementation is given in section 3.6. The penultimate section, 3.7, describes the verification and validation of the model. Section 3.8 concludes the chapter.

3.1   USE OF THE HCSS MODEL

The HCSS model will used to gain insight in the execution of plans, by simulating it. This insight is required because the true performance is determined during the execution of a plan and not during planning. Important performance indicators such as transportation costs and delays have a dynamic aspect (transport involves a multitude of time consuming interactions which are not necessarily deterministic). The model's dynamic nature (it mimics system behavior) provides insight in value and distribution of these (stochastic) variables.

The simulation model will be used to analyze single operator planning (the current situation) in chapter 4 and will be used to evaluate bundled operator planning (the future situation) in chapter 7.

3.2 OVERVIEW OF THE HCSS MODEL

This section provides an overview of the objects and their instances endogenous to the model. This means that the infrastructure used for transportation, the container types transported, the barges transporting containers and the terminals handling containers will be described. The inputs of the model are the calls planned (and executed) and the containers transported, and are a part of the experiments conducted. All of the objects (their concepts) used in the model have been described in chapter 2 and for a formal definition that chapter should be consulted.

The demarcation made in chapter 2 constrains the infrastructure of the ports of Amsterdam, Antwerp, Rotterdam, Nijmegen, Den Bosch, Hengelo and Wansum and the interconnecting infrastructure. This however still leaves us with a large amount of objects in this layer. This is mainly due to the vastness of the model, and the conciseness of the graph used to model the physical layer. The physical layer has been depicted in Figure 15. The physical layer consists of waterway links and nodes and ports. Ports haven't been explicitly mentioned in the previous chapter, they are useful because they provide a geographical clustering of terminals. In order to maintain readability only the main waterways, the most used locks and the seven ports called at are described.

Figure 15 Overview of the physical Infrastructure included in the simulation model[6]

Waterways

There are six main groups of waterways within the physical network (the Meuse, the Waal, the Neder Rhine, the Amsterdam Rhine Canal, the IJsel and the Rhine Scheldt Canal). These main waterways and their tributaries have a total length of over 800 kilometers. These waterways have been depicted with white lines in Figure 15. All of these waterways have upper limits for the size of barge that navigates them. However most of the waterways have limits which are larger than the dimensions of the barges used. There are three exceptions which are all caused by locks in the waterway; these are discussed in the next paragraph.

---

[6] The waterways are the (blue) lines. The main waterways have been marked with circles. Locks have been marked with squares. The ports involved have been marked with (orange) polygons & (green) arrows.

Locks

There are 21 locks within the physical layer that need to be considered (Only the ones outside ports have been depicted using white squares in Figure 15). Most of these locks only cause a delay in travel time. However four of them limit the size of the barges that can pass and limit the height of stacked containers. These locks are of great importance because they add routing and resource constraints to the planning process. The locks in the Twente Canal (IJsel waterway which connects CTT to its destinations, the locks near Eefde and Delden) can only be passed by barges which have a maximum of two containers stacked on top of each other, this effectively reduces the maximum utility rate of most barges (assuming that normally four containers can be stacked on top of each other) to 50%. The lock near Engelen connecting the BCT terminal to the Meuse (and all of its destinations), limits the maximum size of the vessel used to 90 meters in length. The lock near the Weurt in the port of Nijmegen has a height limit of three stacked containers; this limits the maximum capacity of barges travelling between the Meuse and Waal waterways.

Ports

There are seven ports (Amsterdam, Antwerp, Rotterdam, Nijmegen, Den Bosch, Hengelo and Wansum) within the physical infrastructure. All terminals are located within a port. The ports have been depicted using orange polygons in Figure 15. Ports provide the essential infrastructure for a terminal (such as roads, waterways and quays) to function. Of the ports in the system the ports of Antwerp and Rotterdam contain the largest and most terminals, this is due to their function as hubs within the global supply chain. The port of Amsterdam is, from container transport oriented view, a small container port which only has three terminals which account for 1% of the total transported volume. The other four ports contain the terminals managed by BCTN, and always are the on the endpoint of transportation (either in a receiving or sending role).

Figure 16 The Port of Rotterdam[7]

Terminals

There are 38 terminal locations in the port of Rotterdam at which containers are regularly handled. The largest are the ECT Delta Terminals (The same location, but ECT has made a functional division into five different terminals), ECT Euromax, APM-T, ECT Home, RST and Uniport[8]. The location of these terminals has been depicted in Figure 16. Note that the terminals are geographically clustered into two areas: the Maasvlakte (on the left) and the Eem- Waalhaven area (on the right).

There are 24 terminal locations in the port of Antwerp. The largest terminals in the port of Antwerp are the MSC Home Terminal, the ICL Independent Marine Terminal, the Noordzee Terminal, the Europa Terminal, the Antwerp Gateway and the Antwerp International Terminal. Almost all of the largest terminals (except for the Antwerp Gateway) are owned by a joint venture between the Port of Singapore Authority and Hesse Noord Natie (PSA-HNN). The locations of these terminals have been depicted in Figure 17. The port of Antwerp is different from the port of Rotterdam because its western and eastern docks aren't under the influence of the tide (its central docks are), and are separated from the Scheldt river by the largest locks in the world (these are required for the large vessels to pass).

---

[7] The largest terminals have been marked using white arrows, all white areas are terminals.
[8] This has been based on the analysis of three of BCTN's modality instances.

These locks cause significant delays for barges which are visiting both the tide and eastern docks of the port of Antwerp (the western docks are hardly visited by container barges).
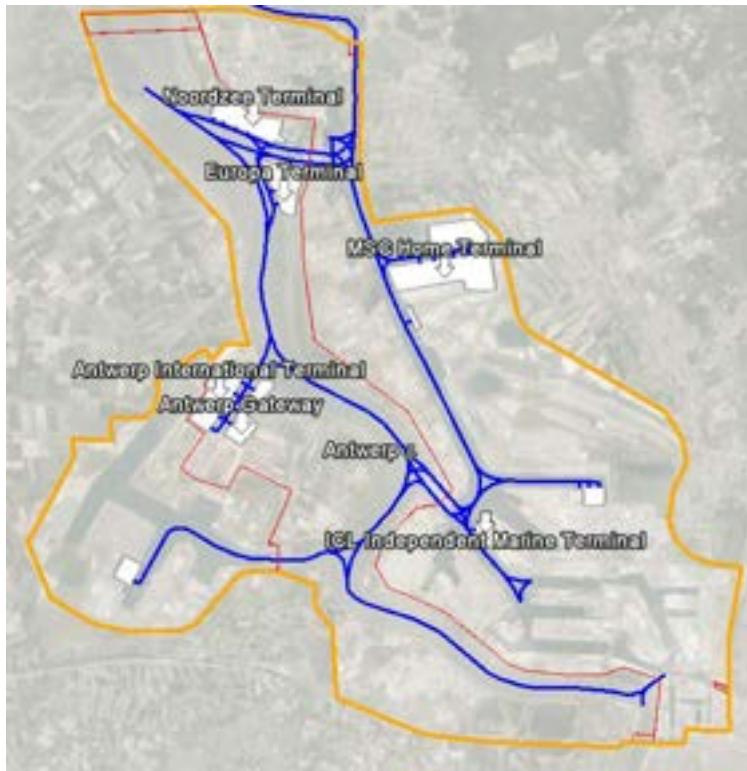
Figure 17 the port of Antwerp[9]

Barges

BCTN hires barges in order to transport containers. Twelve barges of these barges are hired on a regular basis. A barge is usually associated with one of the BCTN terminals in order to reduce planning complexity. The Balance example is associated with the WIT terminal. Figure 18 shows the activity of the Balance between February and April 2009 and how it commutes between Wansum, the ARA-ports and a German port – the intensity of the lines show that the balance mainly commutes between Wansum and Rotterdam. The capacity of the barges range from 90 TEU to 208 TEU. The table below gives an overview of the barges used (This data has been obtained during the analysis of the three Modality instances and has been augmented using data from the SchepenDB: Vereniging de Binnenvaart 2009).

Table 5 Barges used for transportation

Name	Terminal	Size (Length x Width M)	Capacity (TEU)
Caan	BCT	90 M x 11.5 M	99 TEU
Manuela	BCT	80 M x 9.5 M	90 TEU
Nirvana	BCT	90 M x 11.4 M	99 TEU
Colorado	CTN	85 M x 9.5 M	90 TEU
Erculano	CTN	95 M x 9.5 M	99 TEU
Ophir	CTN	110 M x 11.5 M	208 TEU

---

[9] The largest terminals have been marked using white arrows, all white areas are terminals.

Martinique	CTT	110 M x 11.5 M	208 TEU
NauticTrans	CTT	110 M x 11.5 M	208 TEU
Scopus	CTT	95 M x 9.5 M	99 TEU
Balance	WIT	110 M x 11.5 M	208 TEU
Cunado	WIT	110 M x 12.5 M	208 TEU
Traviata	WIT	98 M x 11.4 M	160 TEU

Figure 18 The paths (the thick grey line) travelled by the barge Balance between February and April 2009

The calls, container transports and containers are part of the model as objects. However their constitution drives the model and from that perspective they are exogenous variables and are discussed in section 3.4. However one of the basic properties of container transport is the type of container used. The main distinction in container types comes in length. Containers usually have a length of 20 (42%), 40 (55%) or 45 (3%) feet. The average size per container is 1.62 TEU. Container types have some implications for transport, most barges have an uneven capacity in length (9, 11, 13 TEU), this means that containers of a larger type can have to be used in combination with containers of the smallest type in order to 'fill' the barge. A more important problem is caused by so called high-cube containers which are taller (9'6" instead of 8'6") than a standard form factor container; this can cause problems on height critical routes and with the balance of barges.

3.3 MODELLING ASSUMPTIONS

A simplified version of the conceptual model, as presented in chapter 2, is used to create the model of the HCSS. This was done to reduce the complexity of the simulation model. The simplifications made are considered to strike a good balance between the required functionality and the complexity of the model. In order to make these simplifications the following assumptions have been made:

1. Locks are not modeled in a regular way. A regular lock is a complex queuing system, in which ships are moved from one link to another link using one or more lock chambers. Locks can be modeled like this, however in order to model this more data on the lock operations would be required. Locks have been modeled by defining special lock links (one for each chamber) which

are available during the locks operational hours and have a treatment time distribution defined on them.

2. Currents for waterway links in the network are assumed constant. Waterways especially those under tidal influence can have varying flow speeds. Previous research done by INITI8 indicates that the difference in flow speeds have only a marginal effect on the outcomes of the simulation model. Another issue with modeling this is the lack of data available. A related issue arises is that required aggregation level for the modeling such behavior properly is very low and requires intimate knowledge of modeling flow based systems.

3. Terminal handling is modeled using one object integrating the quay, the cranes and the personnel. This is done because the explicit modeling of the terminals operations is outside the demarcation and does not add value to the results of the simulation.

4. Barges are flawless and do not have malfunctions, they rarely do.

5. Barge routing is time based. Usually the routing of a barge depends on the calls which have to be executed by the barge and the preferences of the barge captain. Only the call sequence will influence the routing, and the route taken will always be the quickest. The shortest path is not necessarily the quickest; the shortest path could contain a number of delaying locks.

6. Barge speed is a given and is the same for all barges. It is not dependent on factors within the model other than the flow speed of the waterway links traversed, locks passed and some variance. In reality the barge speed is dependent on the preference of the captain and the time and distance until the next call. The used barge speed has a major influence on the fuel consumption of the barge and therefore on the transportation costs. The fuel consumption and therefore the fuel costs are known for a barge's regular speed. The relationship between barge speed and fuel consumption is complex and too little information is available on this.

7. The availability of containers is deterministic; this means that the availability of containers will not deviate from the availability information used for creating the plans. This means for example that opening and closing times of containers do not change over time. There is too little information available in order to simulate mutations properly.

3.4   EXOGENOUS VARIABLES

This section introduces the exogenous model variables. Exogenous variables are not influenced by the model, they in fact drive the model. There are three distinct types of exogenous variables: system parameters, input variables and control variables. These types are defined and described in the following subsections.

3.4.1   SYSTEM PARAMETERS

System parameters are exogenous to the model in the sense that they represent factors and functionality which is outside the model's demarcation. The structure of the model lies on top on these factors. System parameters encapsulate the behavior (the result of the function) of the underlying structure (such as a lock or a terminal). System parameters in essence cannot be modified within the context of the model. Treatment time at locks, the time per move during the handling at a terminal and the barge speed are the main system parameters.

System parameters either have to be estimated or can be derived from observations. Some parameters are stochastic and others are deterministic, this all depends on the variable and its use. The 'treatment time at locks' and 'time per move' parameters are stochastic variables which are estimated using observations. The distributions of these parameters have been estimated using Arena's Input Analyzer by Rockwell Software. The theoretical distribution with the best fit on the sampled data is chosen as the distribution to represent that data. The best-fit is determined by the square-error of a distribution, the lower the error the better.

Treatment time at locks

The treatment time at a lock is defined as the time it takes a barge to pass a lock area (for example Figure 19). The treatment time at a lock is dependent on many factors, among these are: the time of day, the size of the lock (length), the decay of the lock, the use of the lock and the number of lock chambers. The treatment time at a lock is estimated using observations from barge visits to locks (this comes from GPS data - see Appendix D for a complete elaboration on this process). These treatment times are modeled using an Erlang, a Gamma, a Normal or a Beta distribution.
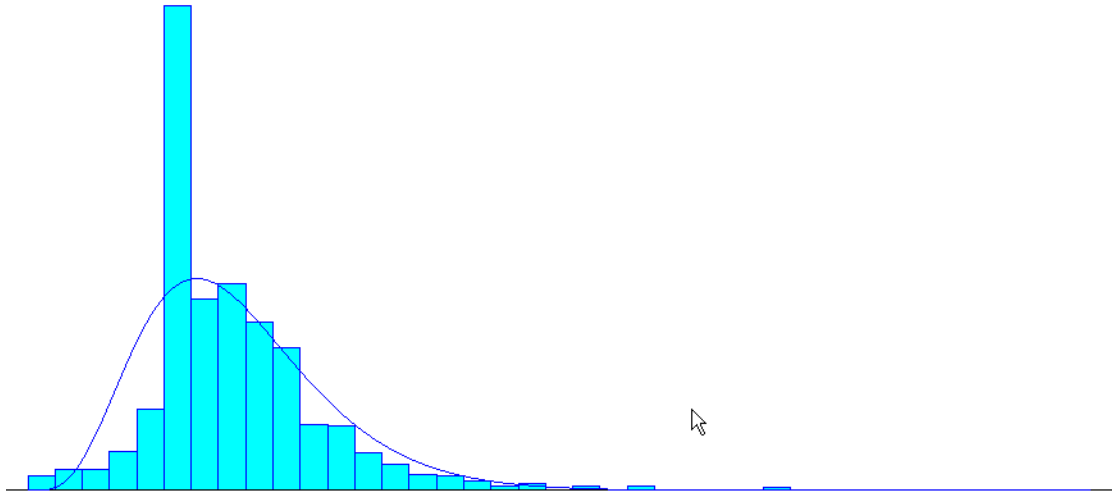
Figure 19 Distribution of treatments time at the Volkerak locks (min/treatment)[10]

Time per move

The time per move during a call is defined as the time it takes to position the crane and perform the move (for example Figure 20). Making observations requires the actual measurement of the duration of separate moves, such data is not available. The time per move is estimated by taking the time a barge spends at a terminal quays (this comes from GPS data) and the number moves made during the call (this data comes from the Modality instances analyze - see Appendix D for a detailed description of this process). For 47 terminals such data is available. Other terminals called at, and terminals which have less than 30 observations will use the overall distribution of time per move.
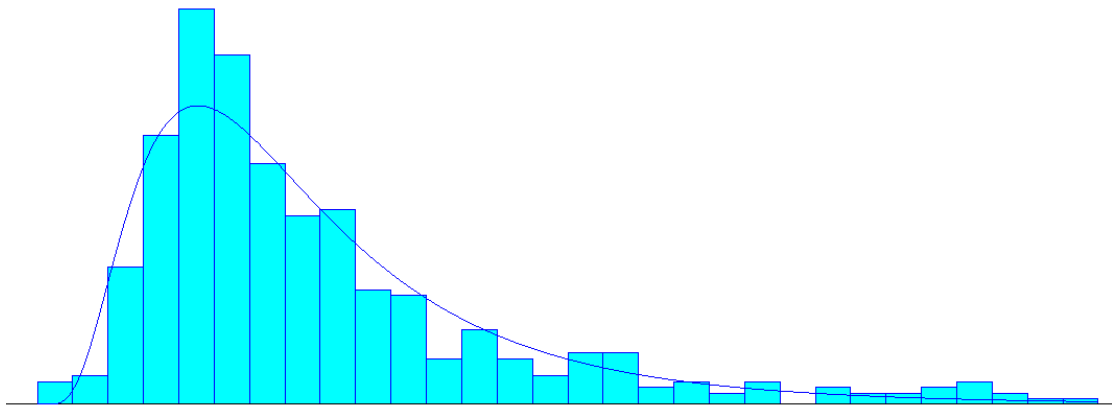
Figure 20 Distribution of time per move (min/move) for the ECT Home terminal[11]

Barge Speed

The barge speed is the speed with which the barge can traverse waterway links. The unit of the barge speed is meters per second. The barge speed has been estimated based on literature (Woltman Elpers 2005; Eggermond 2006; CTT 2008) and fieldwork at the BCTN planning department. Based on this the speed of the barge is modeled as a triangular distribution with its min parameter as 2.7 m/s, its mode parameter as 3.6 m/s and its max parameter as 3.8 m/s.

3.4.2   INPUTS
Inputs are the objects which are transformed by the model into output. Input variables depend on the experiment conducted with the model (the actual input will be described in chapters 4 and 7); only a conceptual description will be provided here.

---

[10] The distribution fitted is an Gamma Distribution ($\beta = 6.63$, $\alpha = 5.23$) with square-error = 0.036161.

[11] The distribution fitted is a Log Normal Distribution ($\mu_l = 8$, $\sigma_l = 5.43$) with square-error = 0.003834.

In this case the main input is the container transport and its properties. The model simulates the execution of these transports from their origins to their destinations. More often than not, input variables are outside of the control of the problem owner and must be seen as a given: container transports and their properties for example are under the control of the shipper and not under the control of the barge operator. The container transport has been described in detail in Chapter 2.

3.4.3   INSTRUMENTS

Instruments control the processes modeled, or they control the parts of the model which are under investigation during the simulation experiments. These are the only variables/objects which can be modified by the problem owner.

In the case of the HCSS model the main control is the call, which is clearly depicted in Figure 13. The call is the solution of the barge operator to the transportation problem the operator is faced with. Calls are appointments for barges to visits terminals at a given time interval during which a number of containers are loaded and discharged. Calls are dependent on each other and have a specific sequence; this sequence is called the route that the barge will take.

The call is constructed during the planning process, which is the focus of this research. This control is what changes during the experiments conducted in chapters 4 and 7.

3.5   ENDOGENOUS VARIABLES

Endogenous variables are internal to the model and dependent on transformations that take place within the model. This section describes the most important endogenous variables. Endogenous variables can be divided into system variables which give insight into the system's internal processes and performance indicators which give insight into the performance of the system. Performance indicators are (almost) always based on system variables.

3.5.1   IMPORTANT SYSTEM VARIABLES

There are many system variables and only the most important system variables will be described here. The following system parameters will be monitored:

Waiting Time

Waiting time is the time an object (in this case a barge) has to wait before it gets treated by the resource they are waiting for (a terminal). Waiting times are unavoidable but they do impose costs. The resource is not utilized while it could be adding value. High waiting times indicate bottlenecks within the system. The only waiting time actively monitored within the model is the waiting time of barges at terminals within the system. This is because terminals can be bottlenecks in the handling of barges.

Treatment Time

Treatment time is the time an objects treatment by a resource takes. Although the resource is actively used during such times, treatment time still imposes costs. The only treatment time monitored within the model is the treatment time of barges at terminals.

3.5.2   PERFORMANCE INDICATORS

The performance indicators monitored within the simulation model are the performance indicators which cannot be directly derived using static analysis of the plans. For all of these variables the exact value is not of great interest. However the way in which these variables are distributed is. The following variables will be analyzed using the model (an overview is presented at the end of this section):

% Containers Delivered Late

Containers should not be delivered late. The percentage of containers delivered late (for which the barge operator is responsible) indicates which fraction of the total number of containers delivered is late. When a container arrives late it may miss a connecting transport by another transport means or cause delays in the clients supply chain. The number of container late is partially influenced by the planning process and the risks taken during the planning process, and can be influenced by events during the transportation process The less risk taken during the planning process, the less influence these events can have. This is a key indicator for the shipper: when the quality of the transport by a barge operator is bad, the shipper will use another way of transport. When containers might be late, barge operators usually resort to trucking to prevent this. The goal of the barge operator is minimize the number of late containers.

The problem with the '*% Containers Delivered Late*' performance indicator is that the indicator requires reliable information on the ultimate delivery time of a container. This information is retrievable from

data, however the reliability of the data is questionable because delivery windows may shift over time (for example due to vessels arriving later or earlier at sea-side terminals); these changes are not always registered in the system. The relative transport duration can be used as an approximation. The assumption is that the longer the transport will take compared to the planning, the more transports will be late. An increase in the relative transport durations is a strong indication that more containers will be late. On its own right it is a good performance indicator for Quality of Service. The relative transport duration can be calculated by taking the transport duration and by dividing it by the minimum transport time required for the transport. Taking the relative transport duration makes transport durations from between different terminals comparable as to Quality of Service.

Barge Utility

The barges utility is the use of the barge's capacity. The more the barge is transporting (at the same time), the more efficient the barge used, the higher the barge's utility is. The utility of a barge is influenced by the availability of containers, their geographical dispersion and the planning process. A barge operator typically wants to maximize the utility of its barges.

The utility of a barge can be a perverse performance indicator due to a number of reasons. The maximum capacity of a barge is influenced by the link it is traversing; this means that the utility can actually change while navigating the network even without any loading or unloading activities. When in a short time period a number of calls are made, during a visit to a port, the utility is greatly influenced by the handling speeds and waiting times at terminal. To prevent such a perverse influence a number steps have been taken, the goal is to measure the barge utility in a more context aware way. This in effect leads to a less controversial approximation of the barge utility. The utility will only be measured while sailing, this reduces the influence of calls on the barge utility. The reference utility is dependent on the path taken, the maximum possible capacity will be used to measure the utility (for example the maximum possible capacity for paths from Hengelo to Rotterdam is 104 TEU for a 208 TEU barge, because of the stacking limitation of two container layers); this prevents that the utility can change during travels. Two distinct ways of calculating the barge utility are possible, one based on the travel time and one based on the travel distance. The travel time based barge utility rate is dependent on the path taken by a barge during the simulation and will change from replication to replication, while the distance based version will not change.

The proposed approach to barge utility measures the use of the barge during sailing. The maximum performance of the barge is dependent on interactions between terminals and barges and on the demand for transportation. The proposed approach shows the barge utility which is under the influence of the barge operator, and is not dependent on capricious interactions.

Deviation from Planned Duration

The deviation from plan shows the difference between planning and execution. The deviation shows the feasibility of the planning process, when deviation is low, planners has shown to be capable of making feasible plans. When the deviation is generally negative, planners are probably creating too conservative plans. The deviation is dependent on the plan and its execution and is generally calculated for activities (calls & hops) by taking the difference between the executed duration and the planned duration. A closely related performance indicator is the delay in execution, which is the difference between the executed and planned end of the activity. The delay in execution of an activity is the sum of the current and previous deviations (assuming that the planned and executed starting times are equal). Since the deviation can be dependent on the delay (for example arriving when a terminal is not opened will force the barge to wait until the terminal opens again) both performance indicators are taken into account.

Costs

Each commercial entity has to make profit or a break-even in order to maintain a sustainable business. Using cost and profit measures in a system that should be used in a competitive multi-actor setting is a rather big ask from the actors involved. Especially the propagation of details on profits made is problematic. Because operators all work with the same costs structures and are subjected to the same cost constraints (fuel prices, barge rents etc), costs are similar. This makes comparing costs less problematic, therefore this research focuses on costs instead of profits.

The overall costs of an operator are a combination of the barges hired, the use of these barges and costs for alternative container transportation. The costs for handling the container and staffing are not taken into account as these costs will not change due to bundling.

An overall cost measure only gives an idea of the size of an operator. The costs should be related to the performance delivered, because this allows for comparing transports for different terminals and routes. For each container transport a certain transportation effort (performance) by the barge operator is required, this is dependent on the size of the container and the distance it needs to transported over. The performance of the barge operator is the total of these efforts. The performance indicator used is the costs per size distance delivered (E / TEU * KM).

A static model has been constructed which is used to calculate the costs. The parameters used within this model are rather generic, but have been derived from BCTN's internal documents (CTT 2008) and interviews. All parameters based on these documents and interviews are deemed business sensitive and will not be published. However the methodology for deriving these parameters is described in Appendix B.

There are two major factors which drive the costs: transportation by barge and transports executed by another modality. Transportation by another modality is generally spoken more expensive than transportation by barge and is only used when transport by barge doesn't meet the transports requirements (a time window or location) or when capacity is not available. The most used alternative modality is transportation by truck. The costs for alternative transportation is calculated by combining the performance (TEU*KM) required by transport and the cost per performance for alternative transportation (E / TEU * KM) parameter. It is assumed that the transportation distance for a container transport by barge is roughly similar to those using an alternative modality. The cost per performance for trucking is used as the cost per performance for alternative transportation, because BCTN has no access to other alternative modes of transports (such as track).

The costs for transportation by barge are a bit more complicated because there are two different types of barge activity: calls and hops. However from a costs perspective both activity types have the same cost elements: barge time and barge travel costs. The barge time is the time that the barge is used for operations (and is the sum of call and hop durations for a given period) by a barge operator, the barge operator rents a barge for a certain price. This price is dependent on the size of the barge and the availability of the barge (some barges are only active during weekdays). The travelled distance costs are specific to hop activities (there is no movement during call activities). The travel costs are dependent on the fuel consumption of the barge, the fuel prices and the distance covered by the barge. The fuel consumption is dependent on the barge size.

Overview

This last sub-section provides an overview of all the performance indicators measured using the simulation model. Table 6 gives a formal overview of the performance indicators used.

Table 6 Performance Indicators used in the Simulation Model

Name	Formula	Unit
% Containers delivered Late	$\dfrac{\#\ containers\ delivered\ late}{\#\ containers\ delivered}$	%
Barge Utility (Time)	$\dfrac{container\ loaded\ size * barge\ time}{capacity\ size * barge\ time}$	%
Barge Utility (Distance)	$\dfrac{container\ loaded\ size * barge\ distance}{capacity\ size * barge\ distance}$	%
Deviation From Plan	$executed\ duration - planned\ duration$	s
Delay	$executed\ end - planned\ end$	s
Costs	$\dfrac{costs\ transport\ by\ barge + costs\ transport\ by\ alternative}{container\ distance * container\ size}$	E/TEU * M

3.6 IMPLEMENTATION OF THE HCSS SIMULATION MODEL

This section describes the formalism used for modeling, the language in which the model has been created and the structure of the model. This section is not an in depth description of the model's

implementation (for a complete description see Appendix C), but is a high level description which allows the reader to get an overview of the model's implementation.

Formalism

The purpose of the model is to provide insight in the performance of hinterland container shipping, and specifically in the performance indicators as defined in the previous section. This requires that containers, barges, terminals, and waterways are modeled. For such a low aggregation level discrete event simulation (DEVS) is the most suitable world view (Zeigler 2003). The choice for a formalism is always influenced by the modeler. However a strong point can be made for DEVS as formalism: one can embed most other formalisms in DEVS, so the choice for DEVS means that a future user can always apply other (more appropriate) formalisms within the model (Vangheluwe and de Lara 2002). An example is process interaction (Nance 1995), which allows for more object-centric simulation.

Language

The Distributed Simulation Object Language (DSOL) simulation suite is used for the simulation (Jacobs 2005). DSOL is a Java based distributed object-oriented simulation suite. DSOL has been developed by TU-Delft, and continues to undergo development. There are a number of reasons for selecting DSOL as simulation language:

1. INITI8 has used DSOL for multiple simulation studies. The investment made in DSOL as a simulation language and the availability of experience with DSOL is important.
2. DSOL is open source and allows for extensions made to it. This easy access to its core makes it ideal for extension; this especially useful within this research, because of the client-server environment DSOL will be used in.

Structure

The structure of the model has been based on previous experience with modeling (similar) shipping systems (Zijpp, Bliemer et al. 1999; Rakt, Melis et al. 2005; Woltman Elpers 2005; Eggermond 2006). For an extensive description of the simulation model see Appendix C. The model takes calls as its input and simulates the execution of these calls. In the current model the calls are read from a dedicated database, and the resulting call executions are written to the database. The data required to set up the experiments is also obtained from the database.

The model itself follows the conceptual layer model as presented in chapter 2. The infrastructure layer is modeled as a network (package nl.initi8.network), using the interface and concepts provided by Java Universal Network Graph library (Madahain, Fisher et al. 2005). A separation of concerns has been made by separating the active, simulated, functionality from the passive, descriptive, functionality of the network objects. This has been done in order to increase the reusability of the network classes. The separation between the active and the passive functionality has been implemented by attaching a (dynamic) context object to the static object (this follows the adage of favoring composition over inheritance (Freeman, Robson et al. 2004)). For example a context which implements the logic required for simulating lock is attached to a lock link.

The model describes how a barge navigates through the network and the interactions during the calling process. During the calling process a barge captain (represented by the BargeAgent) communicates with the terminal operator (QuayAgent).

A Mover class represents a vessel within the model, and implements basic transporting capabilities. A mover is able to move through the network by executing a path from an origin node to a destination node within the network. This path contains a series of successive connecting links which start at an origin and end at a destination. A mover (package nl.initi8.scc.simulation.model) interacts with the context object attached to the network objects it passes (See Figure 54). These interactions determine if the move can pass a link (a move cannot pass when the link is blocked, or when the mover's dimensions are too large for the link) and how much time that will take.

A BargeAgent extends the functionality of a mover. The agent executes the calls assigned to the barge. During the execution of the call (this is including the voyage to the terminal at which the call is executed) the agent monitors the execution of the call (such as start and end operations times, arrival times and waiting times). When the call has been finished all the execution information is propagated to the agent's listeners by means of DSOL's Observer (Freeman, Robson et al. 2004) architecture. The BargeAgent interacts with the QuayAgent for the execution of calls. The barge agent notifies the quay agent of its pending arrival and when the barge is ready for treatment. The quay agent notifies the barge agent when the quay capacity is available, when crane capacity is available and when operations

take place. When the quay agent has enough capacity it will assign one crane and quay space to the barge agent. When the barge agent is moored at the quay and ready for handling, the quay agent's crane commences handling, discharges the containers which need to be discharged and loads the containers which need to loaded. See Figure 53 for a sequence diagram of this process.

Performance

The experiments currently used with the model usually simulate a time window of three days, using a maximum of 13 barges and a maximum of 2200 containers to be transported. The number of events created by such an experiment lie around 30000. Using a machine configured with an Intel Core 2 Duo T5600, 2 GB of RAM memory, Microsoft Windows XP SP2 and Java Hotspot™ Server VM (build 1.6.0_14-b16, mixed mode) about 50 milliseconds to execute a single run. Saving the results of a run to a local SQLServer 2005 database takes much longer, about 700 milliseconds. The overall performance is more than adequate for the current use of model.

3.7 VERIFICATION & VALIDATION OF THE HCSS SIMULATION MODEL

This section verifies and validates the simulation model. The purpose of verification and validation is to check if the model is a proper representation of the conceptual model (internally consistent - verification) and if the model's results are in agreement with reality (validation). Although true verification and validation is impossible, the successful execution of these steps creates trust in the results of the model and the experiments conducted on it (Sterman 2000), and perhaps a better name for this section is 'Trust Building.'

Before delving into the results of the verification and validation it should be noted that parts of the verification and validation phases of the simulation model coincided with the verification and validation phases of the bundling tool. This is because the model and bundling tool use the same domain model (in which the Call class is the main constituent). Errors detected in the logic of the domain classes, had influence on both. Even more said, new insights gained during the creation of the tool were also added to the simulation model. This ensured that the simulation model and bundling tool were well balanced. This section will describe the validation and verification which directly applies to the simulation model.

The validation and verification have been executed using a number of checks and tests. Following the adage as presented in the Mythical Man Month: "*…never take two chronometers to sea. Always take one or three;*"(Brooks 1975, 1995) these checks have significant overlap. The reason why validation and verification are in one section is that most of these checks and tests both validate and verify parts of the model, a separation seems artificial and does contribute to the readability of the report. The tests and checks have been divided into three types:

1. *Syntax*, this checks if the modeling formalism is applied according to the rules.
2. *Semantics*, this checks if the model is meaningful and can be interpreted in its context.
3. *Pragmatics,* this checks if the model represents the relevant elements of reality. This shows if the is useful given the goals and requirements.

The treatment of the model during the validation and verification phases was the same as used during the experimentation, this is described in section 4.3.

Syntax

The syntax of the simulation model has been checked by the environment in which the model was developed. The Java compiler made sure that no syntactical and type (not typing) errors have been made. The virtual machine performs additional checks (for example Null Pointers). More explicit checks have been added to check the state of objects and the prepositions in method calls in order to prevent an illegal model state. Especially these uncovered many errors in a timely fashion. This kind of testing ensures that the model executes and that the model produces results, be them correct or not!

Semantics

The following semantic model aspects have been checked: input variables, model logic and output variables.

The input variables have been checked using the real data. This yielded some strange results. These were found to be due to a mismatch between DSOL's lognormal distribution and the parameters the Arena input analyzer produced. The Arena Input Analyzer presents the parameters for a lognormal distribution in the form of a logarithmic mean and logarithmic standard deviation, DSOL requires the regular mean and standard deviation. By applying conversion formula's (Law and Kelton 1991) this problem was corrected.
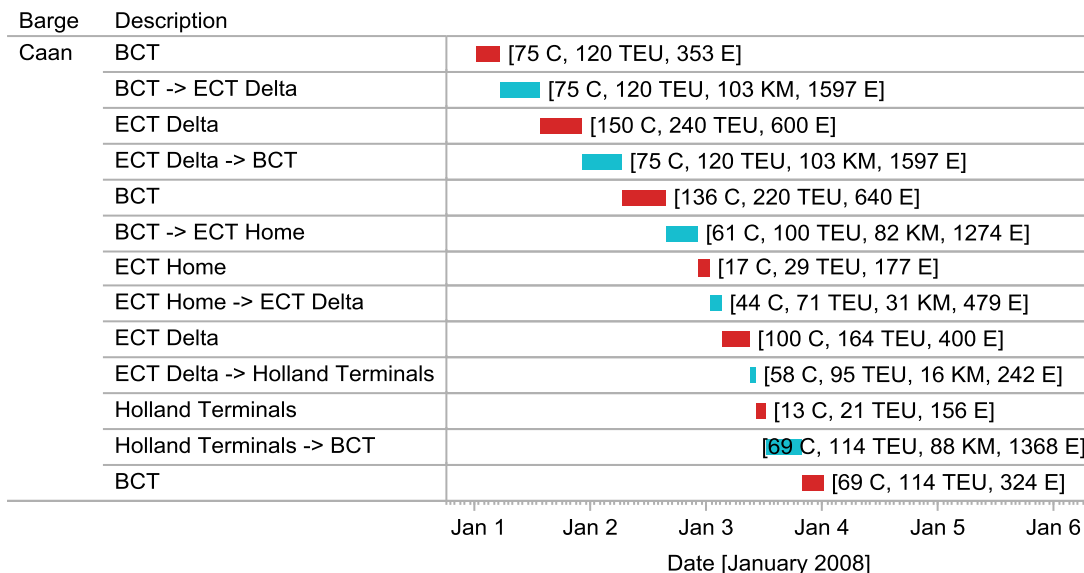
The model logic has been verified in a number of ways. By animating the barges moving across the screen the moving code has been verified. The Java code for the simulation has been reviewed by an expert from INITI8. This yielded some minor improvements. Most of the core components have been checked separately using unit testing (Wikipedia 2009 Unit Testing). A final step in verifying the model logic was analyzing log files of simulations (all active components logged their activities to a logging framework). This step revealed a number of small mistakes:

1. A logic error in the call handling sequence, calls with no required travel (this can happen with initial calls) would not produce valid call execution objects, because of inverted logic. The logic was restructured in order to deal with this.
2. Empty calls (this can happen with initial calls) could lead to invalid state of the barge agent and would block the capacity at the quays. Empty calls should not be executed and executing an empty call violated the assumptions made in the call handling logic. Both the logic for starting calls (Barge Agent) as the logic for handling calls (Barge Agent and Quay Agent) have been modified to handle this corner case properly.
3. Some Barge Agents would stop executing their calls. This actually not a logic error, the logic was sound. This only happened for barges heading for BCT. The call sequence violated the constraints of the model. The maximum depth for the lock near Engelen (which has to be passed in order to reach BCT), was wrongly configured. The value was corrected. Such assumption errors are critical in evaluating the feasibility of plans; the logging of such errors was improved after encountering these problems.

Because the model writes all its output to a database the analysis of output variables was quite straightforward. High waiting times were observed for terminals in the port of Rotterdam and Antwerp. These were traced back to a data entry error: these terminals employ multiple cranes for barge handling instead of one, so the correct number cranes have been assigned. This solved the abnormalities.

The output of the experiments both static and dynamic has been visualized (an example of such a visualization is Figure 21). The simulation model used animation to validate the way barges moved through the system. This confirmed that the barges were executing the calls in the proper sequence and that waiting and calling times were correct. The routing code for the barges was adapted because of observations made, barge take the fastest route, not the shortest route.

Barge	Description	
Caan	BCT	[75 C, 120 TEU, 353 E]
	BCT -> ECT Delta	[75 C, 120 TEU, 103 KM, 1597 E]
	ECT Delta	[150 C, 240 TEU, 600 E]
	ECT Delta -> BCT	[75 C, 120 TEU, 103 KM, 1597 E]
	BCT	[136 C, 220 TEU, 640 E]
	BCT -> ECT Home	[61 C, 100 TEU, 82 KM, 1274 E]
	ECT Home	[17 C, 29 TEU, 177 E]
	ECT Home -> ECT Delta	[44 C, 71 TEU, 31 KM, 479 E]
	ECT Delta	[100 C, 164 TEU, 400 E]
	ECT Delta -> Holland Terminals	[58 C, 95 TEU, 16 KM, 242 E]
	Holland Terminals	[13 C, 21 TEU, 156 E]
	Holland Terminals -> BCT	[69 C, 114 TEU, 88 KM, 1368 E]
	BCT	[69 C, 114 TEU, 324 E]

Jan 1   Jan 2   Jan 3   Jan 4   Jan 5   Jan 6

Date [January 2008]

Type

■ Call   ■ Hop

Figure 21 Planning for the barge Caan

The static plans have been validated by displaying Gantt charts of barge routes (as depicted in Figure 21). These have been compared with Gantt charts of the execution of the plans. This confirmed that the correct barges were used, that they were planned for the correct activities and that the metrics associated with planning were also correct.

Furthermore experts from both INITI8 (Bastiaan van de Rakt & Michael van Eggermond – Simulation Consultants) and BCTN (Bart Spann – Chief Planner) inspected the model's results. They have reviewed the animation of call execution, the Gantt charts of barge routes and data on costs and barge handling for a number of experiments: E[10080], E[51975] and E[102690]. This has lead to some modifications in the parameters for the cost model: The prices for the wrong type of fuel were used; diesel instead of gasoline, gasoline can be up to 30% cheaper. The expert from BCTN also indicated that the waiting times at terminals and locks are influenced by the opening times of these resources. The model does not take this into account explicitly but it does implicitly: by using distributions in which waiting times are in cooperated.

The results in chapters 4 and 7 also confirmed the validity of the simulation model. The behavior of the model was similar to the real behavior of barges.

Pragmatics

The data on planning is already present in the database. During a simulation run a plan is loaded from the database and evaluated a number of times. After every replication the results of call execution are written to the database. The database then contains all the data required to measure all the relevant performance indicators.

The model has been verified and validated. After correcting the mistakes uncovered during the test phase, the model's results can be trusted for the use of evaluating the execution plans for the HCSS.

3.8   CONCLUSION

This chapter described a simulation model of the HCSS which is capable of simulating the execution of plans made for transporting containers by barge. The simulation model has been tested during the verification and validation phases, and is valid and usable for the evaluation of such plans. The model will be used in chapters 4 and 7 to evaluate different planning regimes.

4 ANALYSIS OF THE CURRENT SITUATION

The analysis of the current performance of the HCSS is presented in this chapter. The current costs of planning provides a yard-stick for the evaluation of a bundled planning. The approach is to first create (unbundled) plans per terminal, and then to use the dynamic model to analyze them (see Figure 13). This analysis and synthesis of these results answers the first research question:

1. What are the costs of hinterland container shipping using the current, single operator, planning method?

This chapter defines a number of experiments which have been constructed in such a way that the transport flows and the barge capacity used resemble the current situation. Different planning problems (experiments) of different sizes will be evaluated with respect to the performance indicators defined in this and the previous chapter. During the evaluation of bundling, these experiments will be executed again but then using a bundling planning approach. The differences in costs will make the effects of bundling visible.

Data is needed to analyze the current situation, this data should contain information on barge capacity and container transport flows. The first section discusses how the data has been obtained. The second section shows which variables were retrieved from the data; this is the static analysis of the system (and complements the dynamic analysis as presented in chapter 3). The third section gives an overview of the planning problems selected for experimentation. The validation of the experiments conducted is presented in section four. Section five presents the evaluation of the experiments conducted. The last section, six, draws conclusions based on the evaluation and answers the first research question.

The data presented in this chapter describes BCTN's business process and gives a detailed account of the BCTN's performance. This is business sensitive information and cannot be presented in its natural form. However in order to present a meaningful analysis, the current situation must be described with some accuracy. The experiments presented here *reflect* the current situation (the transport flows and barge capacity used are similar). Not the actual data is used for experimentation, the data used has been generated based on the analysis of the real data. The generation and planning of these experiments is briefly discussed in section 4.

4.1 DATA COLLECTION

The data model used for the analysis of the current situation is the conceptual model as presented in chapter two. This requires that data on container transports, containers, container types, barges, calls, routes, terminals and their quays was obtained from a number of sources. All this data was collected in a central analysis database, a datawarehouse. This datawarehouse is able to store all the required objects. Due to the data warehouse's unique layout, a snowflake design (Navathe and Elmasri 2006), data analysis is quite straightforward. For an extensive description of the datawarehouse see Appendix D.

The first and foremost source for data are the Modality systems that BCTN uses for the registration of their business processes. Data from the modality systems of CTN, BCT and WIT could be used directly, data from CTT's Modality was unavailable at the time of writing and had to be inferred (more on this further on in this section).

Modality is a transactional system, it is designed to efficiently store and update key transactions (like booking, planning and invoicing etc...) occurring in the business process of a barge operator. This is a far cry from the requirements for analysis within this research. In practice this means that there is an impedance mismatch between the data model used by modality and the data model required by this research. A prime example of this is the lack of call registration in modality: modality stores load and discharge voyages in which appointments are stored for loading and discharging containers. A combination of load and discharge (which are part of different voyages) appointments at the same terminal at the same time is a call. Some data is not registered at all because it is not needed for the process, for example the length of the voyages and the execution time of the calls. For this additional source were used.

The data on voyage lengths which is needed to calculate the performance delivered by BCTN came from a map. This map has been created during this research using Google Earth and can be used in a multitude of environments, such as the Java programming language and a SQL database environment. This map provided the path distances between terminals. The data for call execution has been retrieved form GPS data. All regular BCTN barges are equipped with GPS devices. Using the map and the GPS data a barge's visit to a container terminal can be observed, which the execution of a call is. Parameters for the cost model are also stored in the datawarehouse; the fuel consumption and barge rent

parameters have been estimated when constructing the cost model. Others, such as the fuel price came directly from their respective sources (for example data from Transport & Logistiek Nederland (2008) was used to obtain fuel prices).

Due to the different data sources and the impedance mismatch in data, a data transformation phase is required: this is called the Extract, Transform and Load (ETL) phase. The purpose of the extract phase is the extraction of the data from their respective source system. In the case of Modality this was done using a number of custom SQL queries and a physical transfer to INITI8's servers. The next step transforms the data from its source format to its destination format. The data impedance mismatches in the data are resolved and the data from different sources are integrated. This is by far the most labor intensive step during ETL phase. During this step calls (with execution times and cost estimates) and hops are created by combining the different appointments in Modality, GPS data and the cost data. The load phase loads the transformed data into the target database, it makes sure the data is correct and performs integrity checks. The ETL for this research has been implemented using SQL Server Integration Services.

As said no Modality data from CTT was available. Because of the constraints imposed by the Twente Canal all barges coming from CTT have up to 50% of their capacity unused. Both BCTN and INITI8 view this as an easy win in bundling and wanted to be able to evaluate this opportunity. However aggregated data is available for CTT. This data has been used to generate source data representative for CTT's situation in 2008.

4.2 PLANNING VARIABLES
This section describes the variables used for the data analysis. The variables introduced in this section are focused on the creation of the planning, and not on its execution, this is why they can be analyzed statically. There are three types of variables discernable: variables that drive the planning process, variables which control the planning process and variables that are a result of the planning process.

Note that the variables introduced in chapter 3 are also a part of the static analysis. The static analysis provides an ex-ante view on these variables while the results of the simulations provide their ex post counterparts. The difference between these views gives an idea of the randomness in the system and also allows for the improvement of the assumptions used in the static analysis.

4.2.1 INPUT VARIABLES
The input variables drive the planning process. The variables presented in this subsection mainly describe the scope of the planning problem. Actually they are different views on the same input object, the container transport. The influx of container transports is influenced by factors outside of the system, and can only be marginally influenced by factors such as QoS and origins and destinations served.

Container Transports

The number of container transports a barge operator has to handle gives a clear overview of the demand for the operator's transportation capacity. The variable is a good indicator of the complexity of the planning process. The container is the atomic unit used in planning. The more containers, the more ways a planning can be created and the more complex it gets (from an algorithmic point of view).

This variable is also important on a more strategic level. It describes the demand for transportation aggregated over origin and destination regions (for example the demand for transportation between Nijmegen and Rotterdam); this is called a flow. The size and direction of these flows must match the total transportation capacity (not only barges, as the use of barges remains a choice) and its routing. The better the match between demand for and the supply of capacity, the more efficient the transport can be and the more likely it is that the barge operator is running a commercially viable operation.

Container Transport Performance

The container performance is the effort a barge operator has to make in order to deliver containers. This is different from the number of container transports in that this variable expresses the claim (in size and in distance) of the container transport on the transportation resources of the barge operator.

This variable is fairer than the number of container transports when two terminals are compared: they can have the same demand for transportation, but when they have different distances to a destination terminal the terminal closer by can deliver more containers in the same time interval, while their container transport performance would be similar. An example of this are the BCTN terminals CTT and BCT: BCT is half the distance to the Rotterdam terminals as CTT is, so BCT has to deliver twice as many containers as CTT to match CTT's container transport performance.

4.2.2 CONTROL VARIABLES

Control variables are variables that, like input variables and planning constraints, are exogenous to the planning process. Control variables are under the control of the planning actor, and are an artifact of how (in this case) the barge operator runs the planning process. In resource planning there is one main control variable:

Hired Barge Capacity

The hired barge capacity shows how much capacity is hired by a barge operator over a certain amount of time. This is a major cost factor for the barge operator. The operator wants to minimize the amount of capacity hired while still being able to handle the demand for transportation. The trade off made here depends on the availability of barge capacity, the availability of other modes of transport and the price for transport by that mode. The minimization of hired capacity is threatening to the barge owner, who makes money by renting out its barge(s).

It should be noted that this variable is far from continuous. Its value is comprised of the capacity of the barge hired. The minimum increment in the value equals the capacity of the smallest available barge.

4.2.3 OUTPUT VARIABLES

The output variables describe the result of the planning process. The actual result of the planning process is a number of calls for a number of barges and a number of container transports that are transported by another form of transport. The variables presented in this subsection are the most important variables which describe the resulting planning.

Barge Distance

The distance that the barge covers is a measure that gives information about the barge usage. The larger the distance covered the more fuel it consumes, which is a major cost factor in transportation. In recent years environment awareness is a becoming a more and more important topic. The reduction of sailed distances and thus fuel consumption reduces the carbon footprint of container transportation. The barge operator tries to minimize the distance sailed by the barge, while still being able to handle the demand for transportation. This saves money on the short run and improves the operator's carbon footprint on the long run.

% Container Transports by Alternative Transportation

The percentage of containers transports by alternative transportation is the number of containers that had to be transported by another means of transport (while transport by barge was also an option) divided by the total number of transports. This indicates the quality of the planning process, assuming that transportation by barge is more attractive (Planco 2007; CTT 2008). Sometimes alternative transportation is unavoidable, because of the strict requirements by the shipper. A barge operator wants to minimize this. This variable is hard to measure because of two reasons: the registration of such transports is quite different and it is hard to determine if transport by barge were an option.

Container Detour Factor

The container detour factor is the total length of the path travelled by the container divided by the length of the shortest possible path. This is always a number larger than or equal to one, it is an indication of the effectiveness of the transport. The higher the factor the less efficient the transport is, because containers are travelling a greater distance than needed. The detour factor is influenced by the origin and destination region of the container and by the planning process. High detour factors suggest that planning by the barge operator might be improved.

4.2.4 PLANNING VARIABLES OVERVIEW

This last subsection presents an overview of all the variables used to analyze the planning process. Table 7 gives a formal overview of the variables used.

Table 7 Planning Variables

Name	Formula	Unit
Container Transports	$\#Container\ Transports$	Container

| Container Transport Performance	$\sum container\ size * \sum container\ distance$	TEU * M
Barge Capacity Hired	$\sum barge\ capcity$	TEU
Barge Distance	$\sum barge\ distance$	M
% Container Transports by Alternative Transportation	$\dfrac{\#\ Containers\ Transport\ by\ Alternative}{\#\ Containers\ Transported}$	%
Container Detour Factor	$\dfrac{Planned\ Container\ Travel\ Distance}{Minimum\ Container\ Travel\ Distance}$	%

4.3 EXPERIMENTS

This section introduces the experiments conducted. These experiments reflect the current situation. Between the experiments there are differences (for example in parameters), which can be regarded as different snapshots of the current situation; by combining these snapshots a complete image of the current situation is constructed. In this chapter the experiments are used to analyze the current situation. In chapter 7 they will be reused to investigate the feasibility of bundling; the results of this chapter will be used as a reference in that chapter.

The data used for the experiments came from an extensive data analysis of the data of all the BCTN terminals. The data for CTT had to be generated, because only aggregated data was available. The generation of the data is described in Appendix E. The data on container transports, calls and barge capacity between October 2007 and October 2008 has been used to construct the experiments.

Experiments are chosen for a fixed time window, a snapshot of reality. This time window reflects the normal time window used for planning. BCTN uses a planning window of three days, because relatively long travel times and planning windows used by seaside terminals.

The previous section identified container transport requests as the driving factor behind the HCSS. This 'variable' is used as the main factor in selecting the experiments. The number of container transports determines the overall strain on the system, the more containers to transport the more challenging planning and execution gets. Because of the exponential increase in complexity for the planning and execution of calls at the high end of the container transport spectrum a bias towards this end is introduced in the selection of experiments. The distribution of the container transports per time window is depicted in Figure 22. The selected levels of the container transports values per time window are: 1120 (30%), 1365 (50%), 1610 (70%), 1680 (80%), 1750 (90%), 1820 (95%), 2065 (100%) and 2200 (110% - extrapolated scenario).
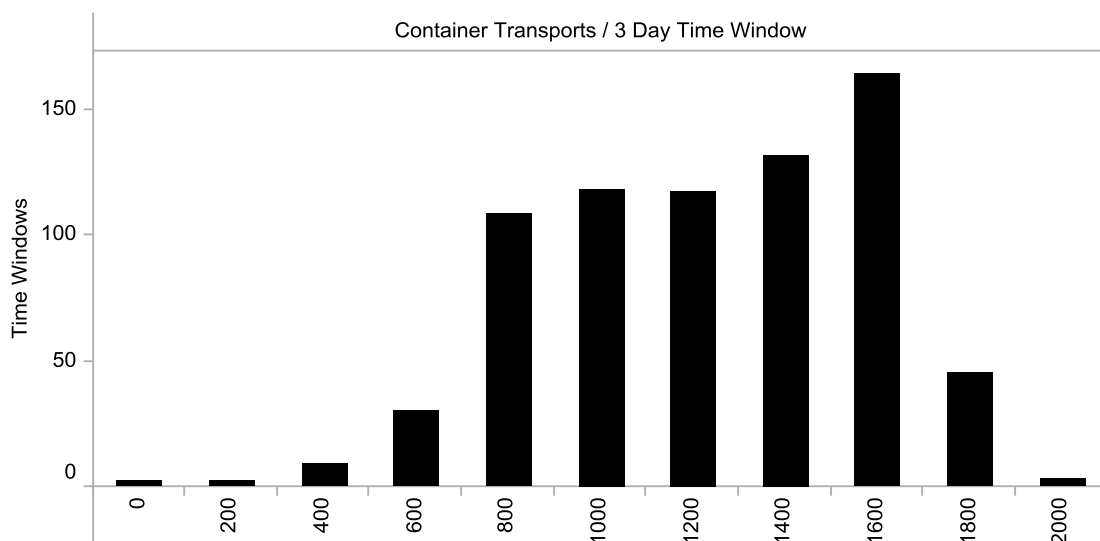
Figure 22 Histogram of Containers per 3 day time window

In order to prevent the selection of an outlier as an experiment, three experiments are selected per level. This leads to a simple experimental design (Law and Kelton 1991) in which a total 24 experiments will be conducted: 3 experiments for each of the 8 levels of container transports selected.

The experiments are evaluated using the simulation model presented in chapter 3. The run-length for each of the experiments is set to infinity, the model will stop when there is nothing left to do (a typical run only contains 30000 events). No warm-up time is required because the model can be initialized in a representative state (the barges are positioned at the correct terminal with the correct loads). In order to create a proper confidence interval for all variables of interest 150 replications are executed per experiment (this is feasible because of the limited calculation time per run: ~50 ms).

4.4 ANALYSIS OF THE CURRENT SITUATION

The experiments selected in the previous section are snapshots of the current situation. This section analyzes each separate snapshot and combines the result of these analyses into one overall picture. The picture constructed helps to gain insight in the costs of the current planning method: it uncovers the value of performance indicators and the total picture gives an idea of the sensitivity of performance indicators with respect to the variables observed. This helps creating a reference to which bundled designs can be compared and allows for identification of bottlenecks and opportunities in the system.

Sections 3.5.2 and 4.2.4 described the performance indicators and other relevant variables. However this analysis does not elaborate on all the variables defined in these sections for several reasons:

1. The variable may be practically immeasurable, this can be caused by the absence of data, errors in data, or data interpretation problems. The *% containers delivered late* indicator is a prime example of this, it is very poorly registered when containers are actually delivered, and changes in time windows are hardly updated in the planning system.
2. The variable does not reveal new or relevant information. This is why the current analysis does not include:
   a. *Container Detour Factor*, this indicator is a too abstract a notion for something which is much easier to comprehend by looking at the costs.
   b. *Barge Utility (Time/Distance)*, this indicator can be easily replaced by costs. This performance indicator has less of the problems associated with the utility indicator, while still showing the system's inefficiencies (high cost values indicate a less than optimal result).

The following two subsections will present the analysis based on the two groups of variables: the first subsection contains the analysis of the variables which describe the planning problems (challenges) encountered. The second subsection presents the analysis of the performance indicators and gives us an idea of the effectiveness of planning and the execution of such a planning with respect to costs and QoS. Note that in order to maintain readability and conciseness these sections will not contain the complete data analysis, but will only present the most relevant insights which add to the understanding of the current performance of the HCSS.

4.4.1 ANALYSIS OF SINGLE OPERATOR PLANNING

The analysis of the plans used within the HCSS has been done by performing a static analysis of the 24 selected experiments. The main goal of this subsection is to show the scope of the planning problems within the experiments and to uncover how the current planning (single operator planning) method allocates resources in order to meet the demand for transportation.

Input

The first step in analyzing planning is to look at the input: the container which needs transportation. Although the *container transports* is the key variable used in defining the experiments, the more informative *container transport performance* variable will be used in analyzing most data. The variable will be mostly used as the independent variable in explaining the value of a dependent variable. This is done because the *container transports* variable does not contain information on the performance which needs to be delivered. This assumes that the two variables are closely related, this is proved by Figure 23 (below).
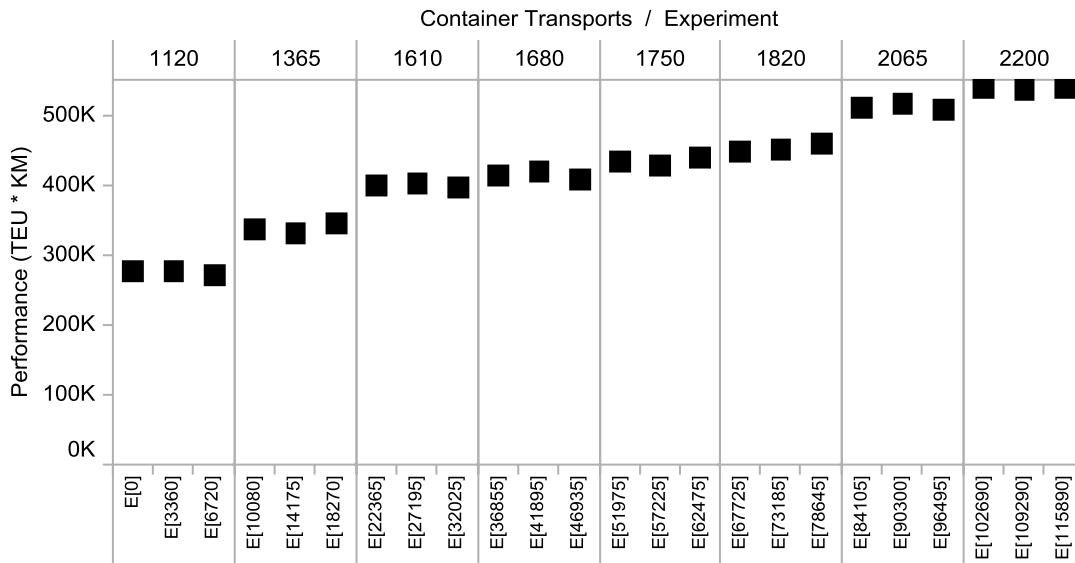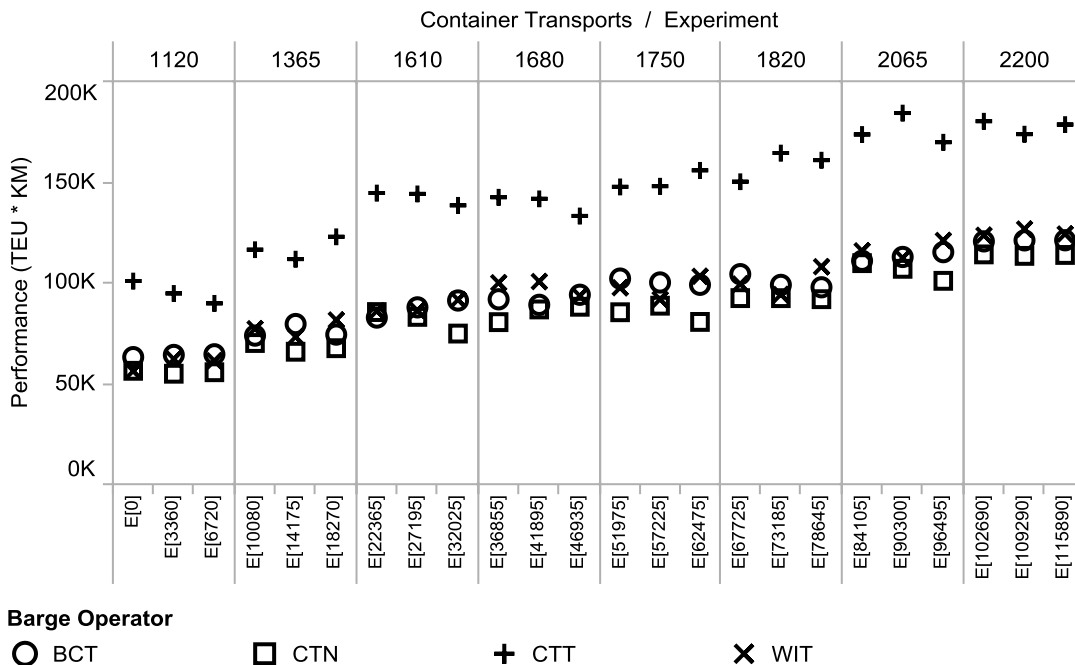
Figure 23 Relation Container Transport Performance and Container Transports

Barge Operator Performance

Four different barge operators (one for each terminal) are taken into account for this analysis. The performance which each barge operator fulfills is an interesting metric, because it gives an indication of the strain which is put on the terminal's transportation resources, this is shown in Figure 24. The major conclusion which can be drawn is that CTT's performance in comparison to the three other operators is much higher (up to 50%). This high performance is mainly caused by the location of CTT's terminal within the network, and its relatively high volumes of transports. The other three operators (BCT, CTN and WIT) have extremely similar performances. This means that the demand for transportation capacity for BCT, CTT and CTN is similar while CTT requires much more capacity.

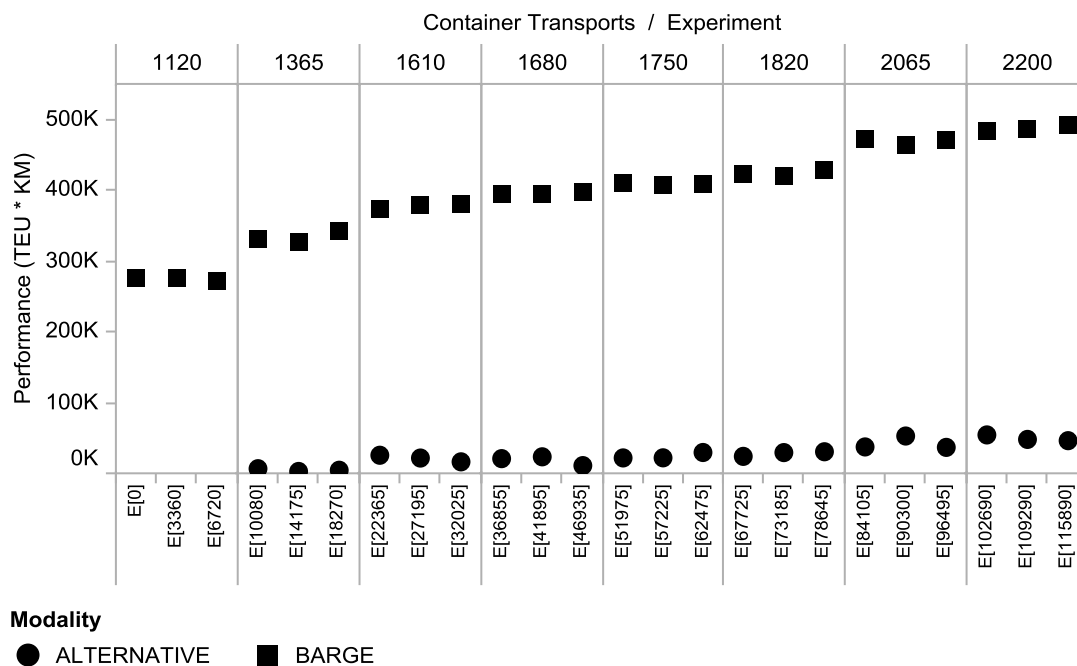**Figure 24 Container Transport Performance per Barge Operator per Experiment**

Alternative Transportation

Alternative modes of transportation are typically only used when the barge operator has no choice (see chapter 3 for a further elaboration). The barge operator's reluctance with using alternative transportation is caused by the higher costs associated with these modes of transportation.

Figure 25 Barge Performance versus Alternative Performance per Experiment

Figure 25 shows this performance transport per barge versus the performance per alternative mode of transportation per conducted experiment. The following observations can be made:

1. Only in the calmest of scenario's no alternative transportation is required.
2. The alternative performance transported increases as the level increases; the performance is relatively stable in the range 1610 to 1820 container transports. The explanation for the latter observation can be that from a capacity point of view the system is robust for this range of transportation demand.

Because the performance in a single operator planning scenario is tested it is interesting to know how the use of alternative transportation is distributed among operators. When analyzing the data, it became clear that the main source for trucking is CTT; while the other terminals hardly require any additional transportation (CTN requires some trucking for 100%+ experiments). This is depicted in Figure 26; the figure clearly shows that alternative transportation (in this case trucking) has a significant role in transportation for CTT and that CTT requires additional transportation almost from the outset (only the experiments at 30% of maximum load do not require additional alternative transportation). This is caused by a number of reasons:

1. The current barge capacity used by CTT is not adequate for the demand for transportation. The current capacity has a performance ceiling around $110000 - 120000$ TEU*KM, beyond this ceiling alternative transportation has to be used.
2. The data for CTT is estimated, the data used for estimation did not separate trucking and barging – so all transports have been planned. This actually validates the method for generating CTT's plans. The other terminals use alternative (trucking) transportation in rare cases, so their numbers are much lower.
3. The network constraint imposed by the Twente Canal cause transportation by barge to be less efficient then it could be. This makes the costs difference between transportation by barge and alternative transportation smaller.

Further analysis of the use of alternative transportation is outside the scope of this research and will not be taken into account. However in order to perform costs analysis standard values have been used to deal with the costs for alternative transportation. The use of these values is kept to a minimum, because they do not give new insights.
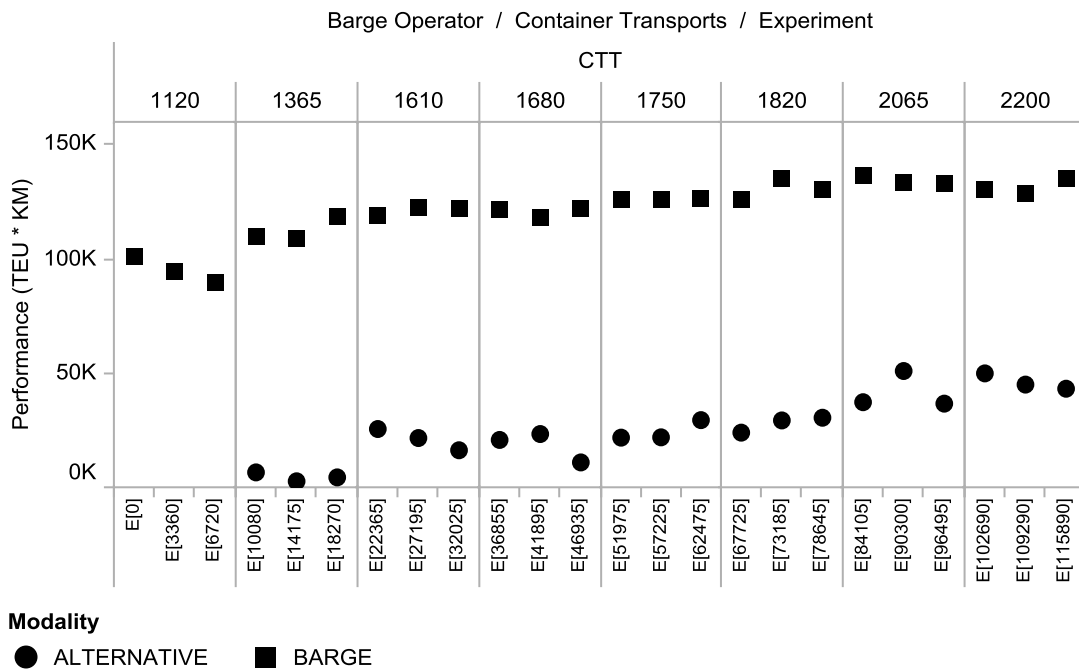
Figure 26 Barge Performance versus Alternative Performance per Experiment for CTT

Barge Capacity

The barge capacity used for transportation is the major cost driver (and sometimes the only one) within the transportation of containers. The use of barge capacity has two aspects to it: the amount of capacity hired (TEU) and the level of activity of the barge. The barge distance has been chosen as an indicator of the level of activity of the barge because this is a stable indicator of the barge's use (unlike time which is dependent on the number of containers the barge transports and on delays within the system).
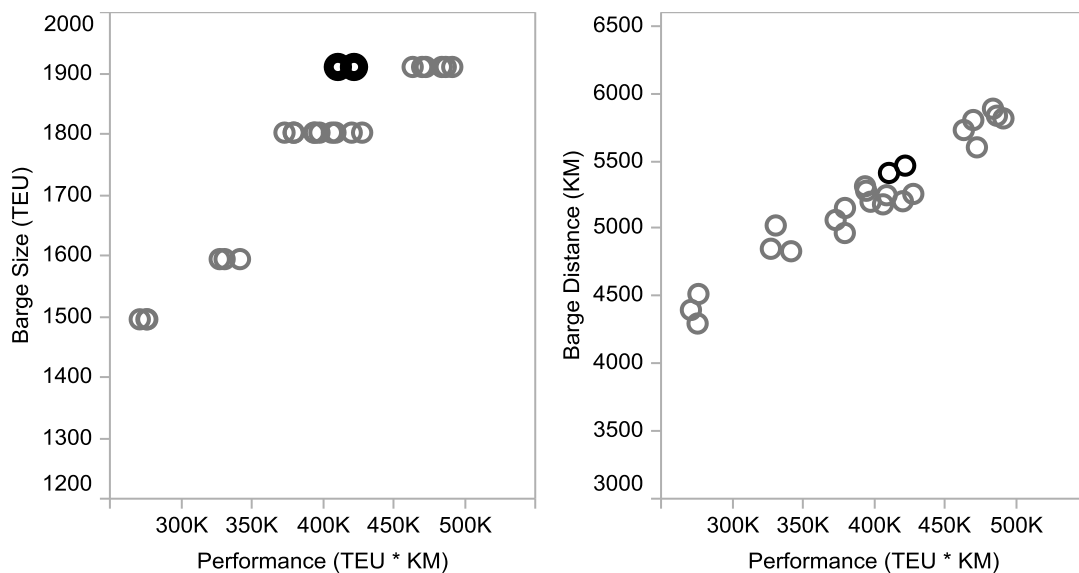
Figure 27 Barge Size and Barge Distance per Performance

Both the size of the barge capacity hired and the barge distance have been depicted in Figure 27. Although rather trivial, it can be (globally) concluded that both the barge size used and the distance travelled used increase as the demand for transportation increases. There are a few observations which can be made:

1. The use of barge capacity increases by barge capacity units; this is clearly seen in the left graph in Figure 27. The implication is that a slight increase in transportation demand can cause a

disproportional increase in capacity, which is unavoidable. In some cases alternative transportation is more attractive, than hiring a new barge.

2. The highlighted experiments show experiments for which a capacity of 1802 TEU was not adequate and 1910 TEU was used, while for similar experiments (with a similar demand for performance) 1802 was adequate. This can be explained by the fact that these highlighted experiments had a higher complexity (caused by for instance different constraints or different flow sizes), and could only be solved with more capacity.

Barge Operator

⃝ BCT	▢ CTN	+ CTT	✕ WIT
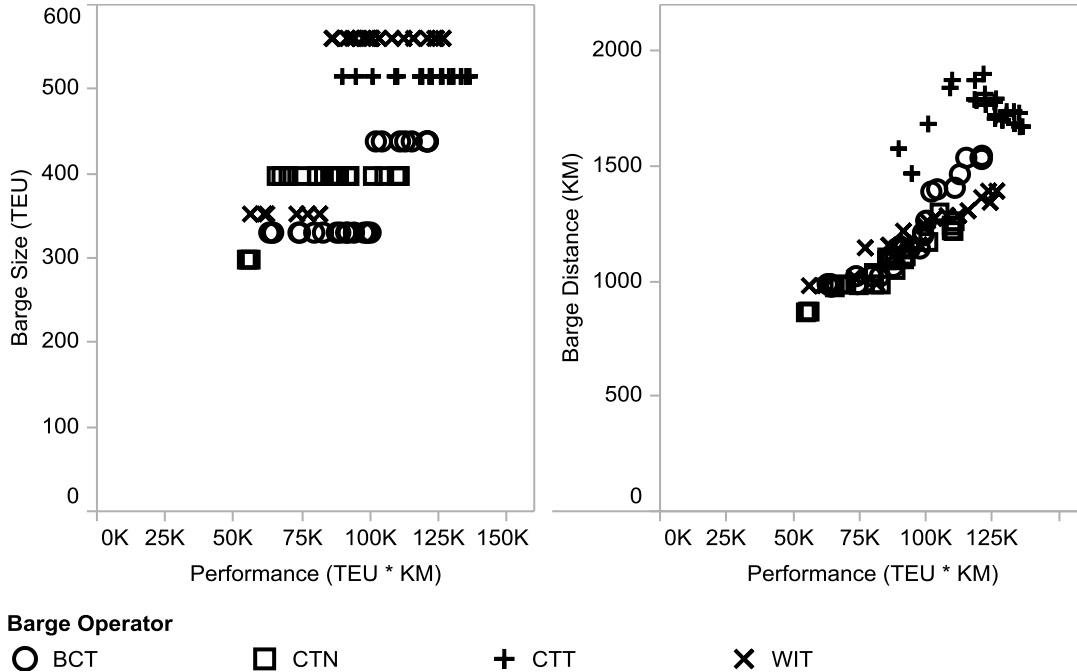
Figure 28 Barge Size and Barge Distance per Performance and Barge Operator

When looking at the use of barge size hired and the distance sailed by the barge at the level of individual barge operators, as depicted in Figure 28, three things become clear.

1. The barge capacity use of CTT is at its maximum (515 TEU) right from the outset. This means that when this capacity is full, CTT has to resort to trucking.
2. The distance travelled by CTT barge actually drops as the demand for transport performance increases. This is caused by the fact that the hired barge capacity can be used more efficiently. Container transports for terminals which would require a detour are handled by trucking.
3. The three other barge operators are a little more flexible than CTT in their resource allocation. It is only a little because they can basically only choose between a low and high level of barge capacity, the increase is exactly one barge (often the most expensive barge to use).

4.4.2   ANALYSIS OF SINGLE OPERATOR PERFORMANCE

This subsection analyzes the effectiveness of planning and the execution of such a planning. The analyses focus on the costs and QoS aspects of performance since these are essential for barge operator in order to maintain sustainable business. Performance analysis involves the analysis of the execution of plans; this has been done using the simulation model.

Costs

The cost indicator allows for the direct comparison of barge operators and planning methodologies. For a barge operator costs are the ultimate performance indicator: almost all other indicators can be translated to costs and it is the primary indicator used for decision making by a barge operator. The barge operator aims to minimize costs.

The analysis focuses does not focus on the overall costs required for transportation, only the costs for barge operations are taken into account. When using for instance container transport performance only

the performance by barge will be taken into account.
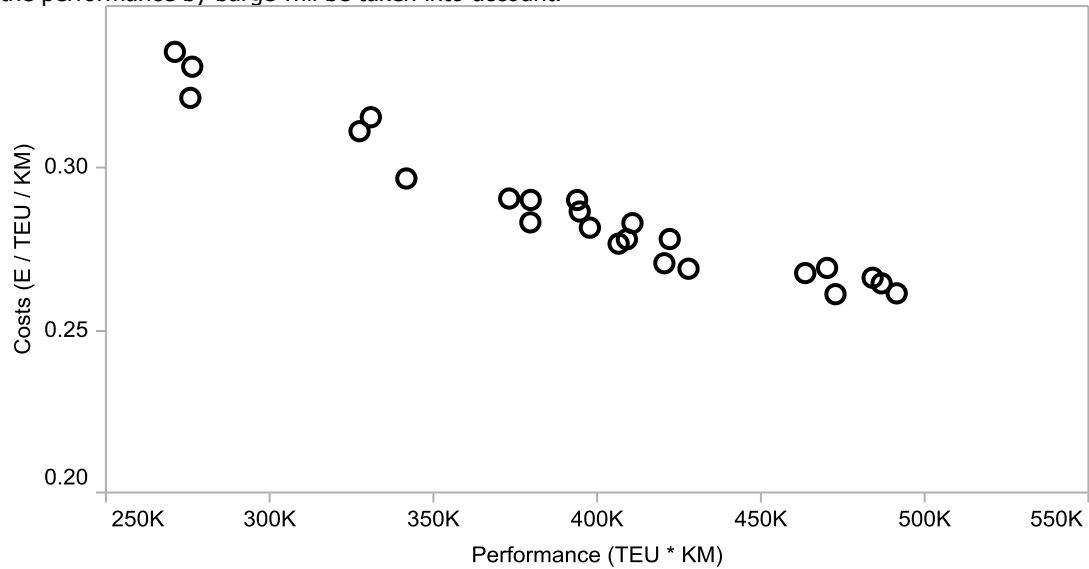
Figure 29 Costs per Performance

The first observation which can be made is that as the performance increases the costs per unit per kilometer drop. This is shown in Figure 29. This is due to the fact that as performance increase barges can be used in a much more efficient way (for example: barges can be optimized for a single route). This is however a global trend and in some instances the costs actually increase. This can be explained by the differences in complexity for each experiment.
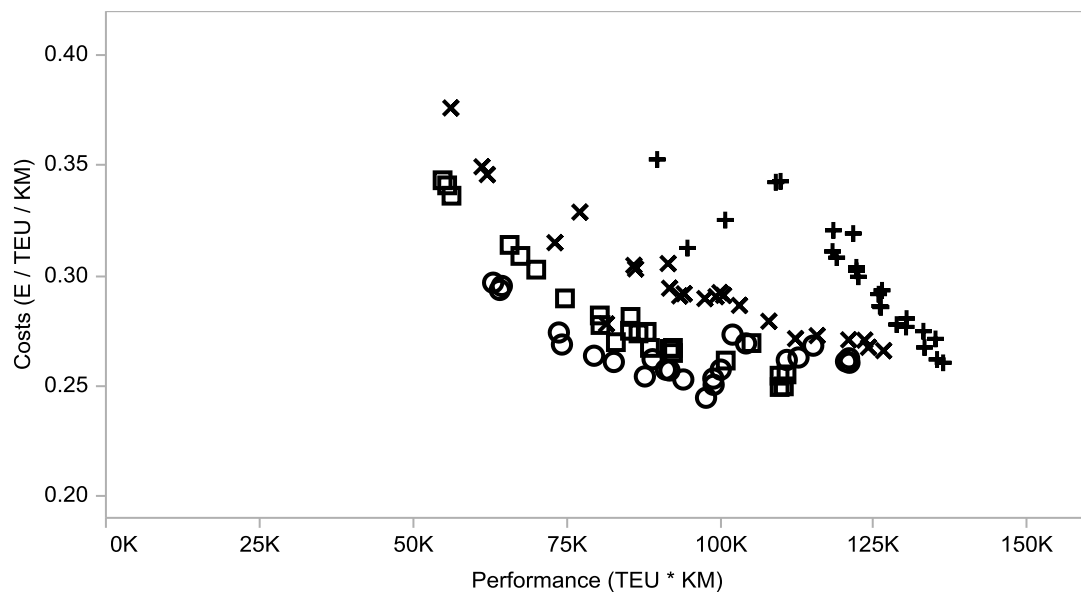
Barge Operator

○ BCT   □ CTN   ✚ CTT   ✕ WIT

Figure 30 Costs per Performance per Barge Operator

The performance per operator shows an interesting pattern, see Figure 30. Globally the same decrease in costs becomes visible. However the costs for each operator are 'layered' around the origin of the graph. For example the costs for BCT are lower and remain lower as those of CTT. The closer an operator's inland terminal is to its opposite terminals (in Rotterdam or Antwerp), the closer the costs are to the origin of the graph. This means that network location, in this case distance, is a major costs driver for barge costs.

Figure 30 also shows the change in costs when the operator increases capacity in order to fulfill the demand for transportation. The decreasing cost trend is broken and is restarted at higher costs. For WIT this happens at 80.000 TEU * KM, and for BCT this happened at 110.00 TEU * KM. The latter capacity increase is interesting because the costs for transportation stay higher than before the capacity increase, while it is expected that the costs should start decreasing again. This is probably caused by the fact that the cheapest capacity is hired before more expensive barge.

The costs analysis so far has been focused on costs estimates made during planning. The actual costs of operations are determined by its execution. The execution of operations has been studied using the simulation model. As shown by Figure 31 the costs of execution are almost always higher than planned costs (> 0%). This means that during the planning phase costs are structurally underestimated. This could be caused by the use of too little slack in creating the plans or the planning or by the coupling (Perrow 1999) of high risk activities (for example the subsequent calls at two unreliable terminals). A PoC must take these risks into account. The current planning approach should use m]ore conservative planning constraints (for example better estimates for treatment and sailing times) in order to create more realistic plans.
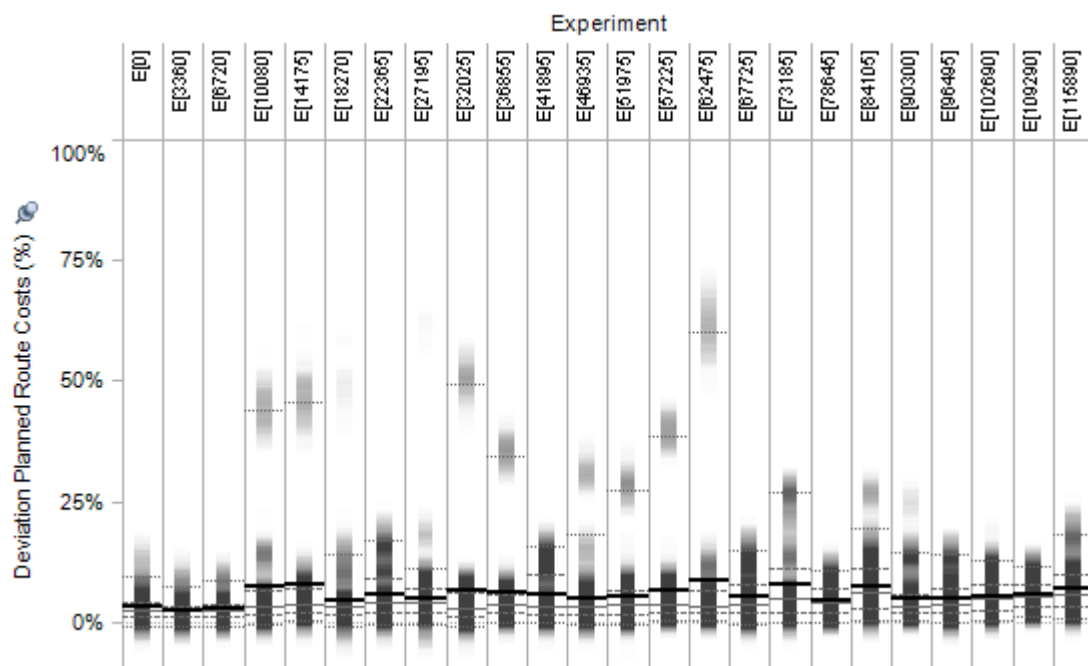
Figure 31 % Deviation from Planned Route Costs per Experiment[12]

What is also interesting to see is that when plans get more complex (more containers), the deviation from planning remains in the same range (with less apparent outliers). This means that the current planning process is robust in the face of added complexity. Bundled operator planning must retain this robustness or improve on it. A decrease in reliability will make bundled operator planning less profitable and jeopardizes the feasibility of bundling because its results will be less usable.

Deviations in handling are not uncommon (these are mainly due to path dependencies in handling). However Figure 31 shows that the deviations are rather high. In extreme cases cost deviations up to +70% are observed. There is an explanation for some of the groups of extreme values (observed in experiment E[62475] for instance, see Figure 32). These are caused by the attempted use of already taken terminal capacity (inland locations can only handle one barge at a time), so the barge has to wait until the capacity becomes available. This is caused by planning in which the one of the effects of barge planning, the use terminal capacity, is not considered. This can be avoided, but currently produces infeasible plans. Bundled Operator Planning has to deal with such effects because it should not create even less reliable plans.

---

[12] The horizontal markers depict the statistical properties of the distribution per experiment (thick black marker = average, thin grey marker = median, thin dashed grey marker = 25% or 75% percentile, thin grey dotted marker = 5% or 95% percentile).

Description	Date	
CTN	1/1/2008 10:39:01 AM	▨ Colorado
	1/1/2008 7:40:21 PM	▨ Ophir
	1/1/2008 8:13:13 PM	▨ Erculano
	1/3/2008 6:39:47 AM	▨ Colorado

Jan 2   Jan 3   Jan 4

Date [January 2008]
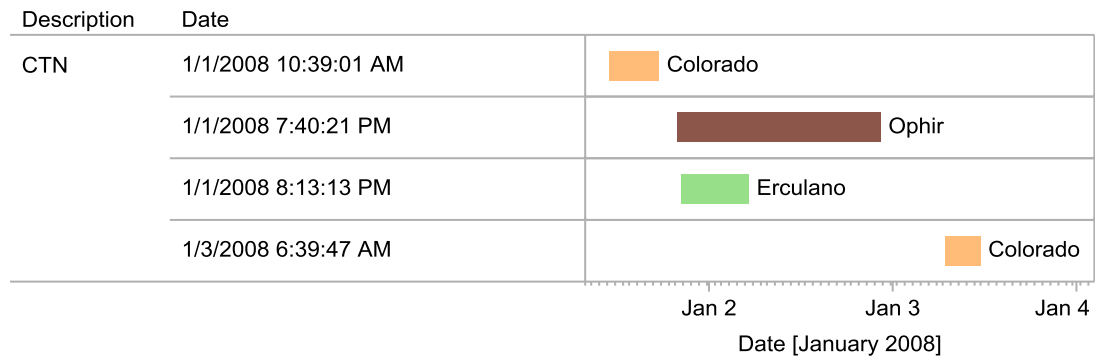
Figure 32 Infeasible plan due to planned overlap in terminal capacity use

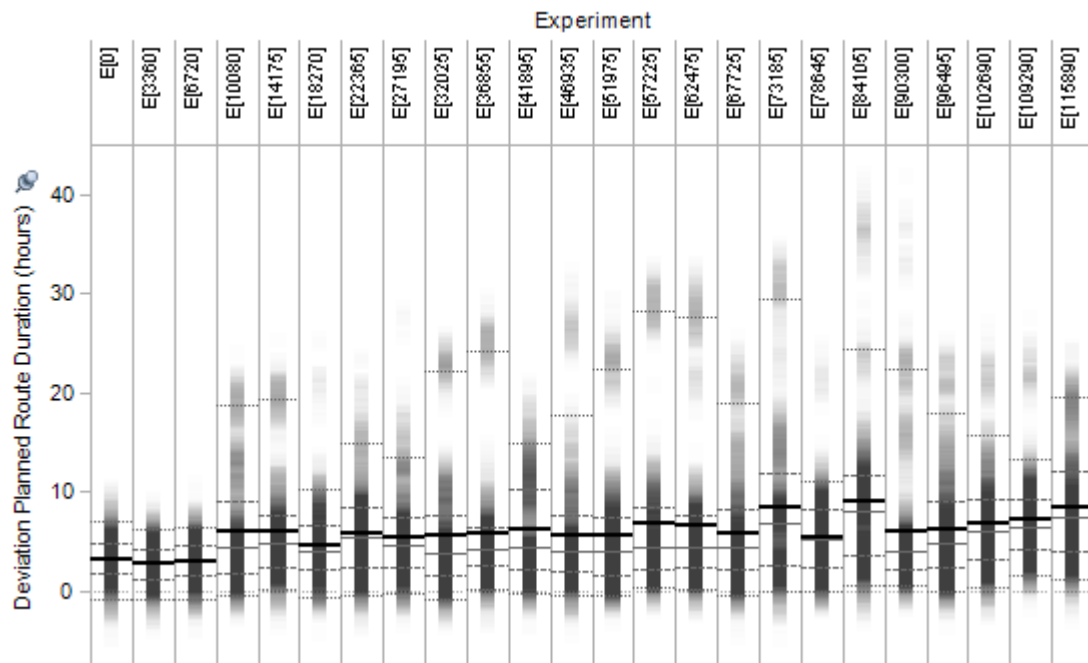The last observation to made is that the deviation in costs is mainly driven by the deviation in duration (the longer the barge is hired the more expensive the transport gets). This result can be observed by noting the similarity of Figure 31 and Figure 33.

Figure 33 Deviation from Planned Route Duration (hours) per Experiment

The costs as discussed here consist of only the direct costs for the barge operator; the deviation in costs with respect to the current plan is taken into account. There are however indirect costs as well: when a barge is late, it is not available for further transportation assignments. These costs are not taking in account in Figure 31. This means the barge operator has to hire additional capacity in order fulfill the demand for transportation.

Quality of Service

The costs and the quality of the transportation services are the main indicators in evaluating the barge operator's performance by its customer, the shipper. The main QoS indicator is the timely delivery of transports. Since this is extremely hard to measure (see the beginning of this subsection and the discussion in 3.5.2), a replacement is found in the deviation in transport duration per container. The deviation in transport duration is the time in which a container is transported from its origin to its destination, it expresses how big delay of the transport is compared to the promised delay time. The larger the delay, the worse the QoS of the transport.

Looking at Figure 34 it can be seen that the average deviation from the planned transport duration slightly increases when the system load increases. The deviations are large (in extreme cases containers

are delivered more than 15 hours late). However their occurrence is hardly related to the size of the experiments (system load), only the smallest experiments have no extremes in deviation. In all cases 95% of the containers are delivered within 12 hours of the planned duration (the top dotted gray markers in the figure). These delays are caused by the handling and planning of barges and have been discussed in the subsection on costs. The number and impact of the outliers within the system seems high (in the view of the author), BCTN should investigate if the transport duration for containers can be improved in order to improve their QoS. This is however outside the scope of this research.
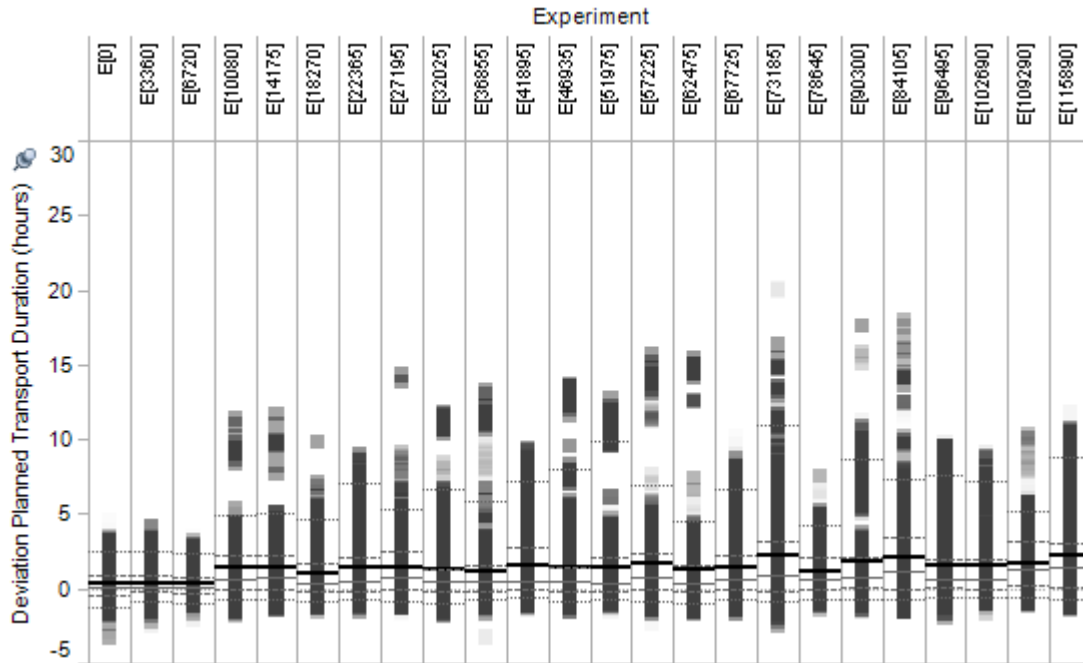
Figure 34 Deviation from Planned Transport Duration (hours) per Experiment

It was expected that the transport duration would be very susceptible to an increase in business. The lack thereof shows stability in the current way of planning. This stability must be retained or improved upon when implementing by a bundled operator planning.

4.5   CONCLUSION

The current approach to planning in the HCSS, single operator planning, and the (resulting) current performance has been analyzed in this chapter. This was required to answer the first research question:

1.   *"What are the costs of hinterland container shipping using the current, single operator, planning method?"*

The purpose of this first research question was to understand the costs of the current planning method and to set a goal for a (future) bundled operator planning approach. The results of the future planning method should be an improvement of the current results for the planning method to be effective.

The analysis focused on both the generation and execution of the plans. The main conclusions are listed below:

1.   The performance of CTT is much higher than the performance of the other barge operators (BCT< CTN and WIT). This is mainly due to the location of the CTT terminal and its relatively high transportation volumes.
2.   The analysis showed that similar experiments can have different complexity. The result is that in situations of equal business more capacity may be needed, in order to fulfill the demand for transportation.
3.   The increase of used barge capacity is dependent on the size of the barges available. A small increase in demand can cause a disproportional increase in capacity. This is unavoidable. However this is tempered by the use of alternative transportation (this can be cheaper in edge cases) and the reduction of overall barge distance.
4.   The use of alternative transportation increases steadily as the levels increase. The source for alternative transportation (mainly by truck) is CTT. CTT's barges have a maximum performance of

110000 to 120000 TEU*KM (depending on the complexity of the planning problem). CTT does not currently allocate more barge capacity than the 515 TEU provided by the four barges it uses. The additional demand for performance is handled by truck. The ceiling in CTT's barge performance is mainly caused by the limitation imposed by the Twente Canal (Only 50% of the barge capacity can be used) and its distance to the terminals in the ports of Rotterdam and Antwerp. These limitations lead to smaller costs difference between transportation by barge and alternative transportation. This difference is smaller than for the other terminals, this means that the advantage of transportation by barge is less apparent for CTT. An interesting side note is that the use of barges (barge distance) by CTT actually drops as performance increases. This is caused by the fact that 'complex' transports are done by alternative transportation whereas barges are used for bulk streams. A recommendation for CTT is to investigate if the use of more barge capacity will improve the efficiency (the cost-quality level) of their transportation processes.

5. The other three barge operators (BCT, CTN, and WIT) hardly required any alternative transportation. Only CTN requires some in extreme cases (> 100%). These operators have the luxury of having a better match between demand for transportation and the supply of barge capacity, be it (usually) only one barge (they always use 3 barges and can switch to 4 barges).

6. The costs per kilometer per standard unit (1 TEU) are decreasing steadily as the levels and transport performance used in the experiments increases. This is caused by the fact that barges can be planned in a much more efficient way when large volumes of containers belonging to similar flows are available for transportation. When analyzing the costs per operator it became apparent that distance plays a major role in the costs; the costs for an operator can be characterized as a function of the demand for transportation and the distance of the operator's home terminal to the ports of Antwerp and Rotterdam (BCT's costs were much lower than CTT's).

7. The execution of the plans yielded that in most cases the costs per kilometer per standard unit is underestimated and that the overall deviation in costs is high (up to 70%). The deviation in costs is stable across the experiments and is not dependent on the experiment's complexity. The deviations in costs are caused by the use of too little slack in creating the plans or the planning or by the coupling of high risk activities. It is recommended that during the planning process more realistic planning assumptions (more realistic durations for treatment and sailing) are used. Bundled operator planning should create plans with the same or less deviation than the current planning approach, otherwise this would increase the realized costs of bundling and would damage trust required for the implementation of bundling.

8. The duration of transports (which is a measure for QoS) increases as the size and the performance of the experiment's increase. The deviations in the duration of transports are high; it can only be guaranteed that 95% of the transports are delivered less than 12 hours late. Even more extreme values are not uncommon. BCTN should investigate what the source of these values is, because a bad QoS will eventually hurt BCTN's business. The measures suggested in the previous point can reduce these extreme deviations. Bundled operator planning should not make deviations in duration worse.

The next step in the research will be to delve into the theoretical concept of bundling and to apply it to HCSS as described in the previous four chapters. Bundled operator planning will change the planning process; the resource allocation within the HCSS will be changed. The following changes are expected:

1. Bundling enables the use of capacity across operators for the transport of their joint containers. This means that as the demand for transportation increases the allocation can be more efficient; the same capacity is hired but this is done in smaller increments.

2. By bundling the need for alternative transportation will be less. The capacity of barges can be reallocated to transport CTT's transports. This will probably not influence the overall costs per kilometer per standard unit, but will improve the overall costs.

5   BUNDLING, AN OVERVIEW

The previous chapters introduced the research problem, the HCSS and showed the current performance within the HCSS. The next steps in the research are to apply bundling to the HCSS and to see what effects bundled operator planning will have on the HCSS. Before the next steps are taken, it is important to know what bundling exactly is. Therefore this chapter gives an overview. The main research methods used for this chapter are literature review and interviews.

Bundling was briefly discussed in chapter one. This chapter will expand on the notion of bundling in the following sections: Section 5.1 approaches bundling from a theoretical point of view and identifies the relevant elements in a bundling problem. Section 5.2 discusses the effects of bundling. Section 5.3 discusses the bundling stereotypes. Section 5.4 shows the ways to implement bundling. Existing cases of bundling are discussed in the last section, 5.5.

5.1   THE CONCEPT OF BUNDLING

This section explores the concept of bundling. By reviewing literature a number of overlapping definitions have been found. These definitions serve as a basis for exploring the concept of bundling. By identifying the elements of bundling, be them physical or otherwise, the concept of bundling is illustrated. The last step will show how these elements are take part in bundling.

5.1.1   BUNDLING DEFINITIONS

Bundling is a specific type of consolidation. Consolidation, according to Binsbergen en Visser (2001), is the combination[13] of transport flows in space and time. Consolidation can be achieved by consolidating in time, in route, in place, terminals and in activity. The purpose of bundling is to increase the utilization of system as a whole and to decrease the use of resources. The definition of Konings (2006) shows the elements of consolidation:

Bundling can be defined as the common transport of freight belonging to different transport relations in common transport units (e.g. barge vessels) and/or loading units (e.g. containers) during common parts of routes. Basically the intention of bundling is to improve the cost-quality level of transport services, but the characteristics of transport flows (volume and spatial pattern) set the pre-conditions, and there will be a trade-off between advantages and disadvantages.

Bundling is route consolidation focused at the transportation resources, but all other consolidation elements can to some extent be found in this definition: this depends on the bundling network design used. The ideas of consolidation and bundling are as old as transportation itself and their applications remain to be one of the core abilities required for a transport operator (Kreutzberger 2008).

Note that in the definitions given Kreutzberger (2008), Binsbergen and Visser (2001) do not explicitly state that different actors need to be involved in bundling, whereas the definition of Konings (2006) does. Within this thesis bundling can also be applied without the involvement of different actors, i.e. transport relations. Different actors are a possibility which add considerable complexity to bundling, but are not required.

5.1.2   BUNDLING ELEMENTS

By reviewing a number of (related) bundling definitions a number of elements have been uncovered. Others can be inferred from the ideas presented in these definitions. The following elements are required in bundling:

Network

Although not explicitly mentioned in the definitions of both Konings (2006) and Binsbergen and Visser (2001), the network is a key element in bundling. Within a network the origin and destination of freight are nodes. The network connects these nodes and enables the physical transportation between the two, when no connection is available between the nodes the network cannot be used for the transportation of the given freight. A network consists of nodes and connecting links; every link connects two nodes. Network links and nodes are resources which have dimensions, a capacity and other properties associated with them. A transportation resource must match these dimensions and other properties of the links and nodes it is using. The capacity of nodes and links constrain the number of transportation resources which can use these at the same time. The usage of a network resource always carries some risk, the two most common risk factors are delays caused by the resource and the downtime of the resource.

---

[13] Their definition actually uses the term bundling here, this is omitted in order to prevent confusion.

Many transportation networks exist. What needs to be noted that there is always some degree of connectivity between these networks and that a transfer between two networks can be achieved by changing transport mode. The network demarcation and the properties of the network elements define the configuration of the network and its scope. The network configuration determines the types of bundling possible. The (re-)configuration of a network can require high investments by the transporting parties; this is because the nodes and links will have to be configured for their role in the network.

Freight

The basis of transport is formed by the need for objects, freight, to be physically moved from one location, its origin, to another location, its destination. The freight has a number of physical properties such as the dimensions, volume, weight and type. The client sets a number of QoS requirements such as service levels and availability and arrival time. The combination of these properties and requirements constrain the transport resource (and its type) suitable for transport of the freight.

Note that freight is a relative notion which has a hierarchical aspect to it. Freight can be combined in a resource, that resource is then transported by another resource, which makes it freight in its own right. Container transport for example is based on this idea.

Freight moving from an origin node to a destination node in a network can be regarded as a flow from the origin node to its destination node over the course of time. The total of these flows represent the demand for transportation capacity. The demand flows cannot be modified, but can be latent. Latent demand is driven by a number of factors: an unfulfilled demand for transportation, new markets and the replaceability of current transportation.

Route

A route is a series of links and nodes that freight traverses in order to move from its origin to its destination. During such a route the freight can change from one transportation resource to another, this requires a transshipment node. The need for transshipment depends on the scope of and the way the network is configured, transshipment adds complexity and costs to the transportation process. Numerous potential routes are usually available within a network. The transportation resources selected must be able to traverse their respective part of the route. The route must also respect the quality of service demands for the freight transport. The chosen route depends on the preferences of the routing entities; it does not have to be the best route, but is one of many possible routes.

In order to improve the cost-quality level of transport, bundling places freight onto shared transportation resources. This can only be done when parts of these routes (trips) can be combined (in space and time) or can be modified in such a way that they are combinable. Bundling will change the way freight is routed in a network, and thus changes the chosen routes.

Resources

Resources transport freight within the network using (a part of) the route given. A resource has dimensional, performance, and capacity properties. These must match the network it is navigating and the freight it is hauling. Transport resources are scarce and their operators try to route (by means of hops and stops) these resources in such a way that they meet the demand for transportation capacity in such a way that the resources are used as cost-effectively as possible. This in itself is one of the reasons why bundling is appealing for transport operators. There is one key requirement in using a resource for bundling and that is that it must be able to hold more than one piece of freight.

The transportation resources available from one node in the network to another node in the network can also be seen as a flow, the total of these flows represent the supply of transportation capacity. The supply flows within the network can be modified by transport resource operators. Such modification entails the choice for transportation resource and the routing (the flow) of the transportation resources. The transport operators try to route their resources in such a way that they are able to meet the demand for (latent) transportation.

Stop and Hop

The resource typically navigates through the network and stops at transshipment nodes. During these stops nodes freight is loaded and/or unloaded. The movement between two subsequent stops in called a hop. A hop is usually the best path possible between the two stops.

The resource operator is responsible for planning these hops and stops. The way the resource is scheduled is of major influence on the costs for transportation (hops and stops are activities which cost

money). The operator tries to create the most efficient plans for its resource, during which the most freight is transported for the lowest possible costs.

Trip

Trips are the resources' contributions to the chosen route for freight. The trip starts at the stop where the freight is loaded on the resource and ends at the stop where the freight is unloaded from the resource. The load and unload stops do not have to be subsequent; there can be intermediate stops (and hops). The trip of the freight loaded on the resource depends on the other freight loaded on the resource which needs to be loaded or unloaded at one of the intermediate stops; this is ultimately determined by the plan created by the resource operator. This (usually) means freight is not taking the best path between load and unload stops, and takes a detour. Detours actually occur both on the trip level and on the route level. Detours are a major source of costs while bundling freight on a resource.

Actor (Transport Relation)

There are different actors which can be involved in bundling. These can have a number of different roles, which can be combined into one actor. The following roles are relevant to bundling:

1. The freight owner, who needs the freight transported from its origin to its destination.
2. The freight shipper, who takes care of the transportation of the freight. The freight owner is a client of the freight shipper; the owner gives assignments of freight to transport from their origins to their destinations.
3. The transportation resource operator, the operator's resources transport freight between two stops in the network. The shipper is a client of the operator and gives the operator assignments of freight to transport between two nodes in the network.
4. The transshipment node operator, the node operator takes care of the transshipment of freight from a transport resource to another transport resource. When the freight is changing modality transshipment is required. The node operator also provides some storage for freight which is not immediately transshipped to its connecting transport. The shipper is a client of the transshipment node operator and gives the operator assignments of freight to transship.

In the definition of Konings (2006) the transport relation is the transportation resource operator. This definition is focused at a horizontal cooperation model. Bundling in the view of the author can range from single actor setups in which it is a mere optimization effort to multi actor and multi role setups in which these actors coordinate on a supply chain level.

In order to bundle freight flows, the resource planning of multiple actors needs to be coordinated in order to create a feasible joined resource plan. This joined plan must be an improvement over the individual resource plans. It is this coordination what makes bundling hard, because of direct coordination costs (for example the act of communication) and indirect coordination costs (for example the abuse of shared information).

5.1.3   BUNDLING DEFINITION IN THIS THESIS

This subsection is concludes with the definition of bundling which will be used in this thesis. This definition is a synthesis of the definitions as presented in the first subsection and the elements as presented in the subsections. The goal is not to replace the existing definition, but to have a clear and concise definition to base further work in this research on.

The first premise of bundling is that it requires multiple items of freight which have their own set of properties and QoS constraint and can be owned by multiple freight owners which need to be transported from an origin node to a destination node within a network. The origin and destination node must be connected.

The transportation of this freight is handled by a shipper. The shipper determines how the freight is routed through the network. This is very dependent of the network structure. The shipper determines which modalities are used for transportation of the freight and at which nodes (if any) the freight will be transshipped. The result of this is a plan for a route with a number of subsequent trips. The challenge for the shipper lies in creation of a cost effective route which meets the QoS requirements set by its client.

The transshipment of freight is usually only applied by a shipper when the origin and destination nodes are not reachable by the same mode of transport and more importantly when freight is consolidated (bundled) in to larger batches in order to organize more efficient transport. The consolidation of freight

in intermediate nodes is a pre-condition for bundling. This freight can be under the control of different shippers. However the freight should have matching trips with similar QoS constraints.

The shipper assigns the execution of a trip to a transport resource operator. The resource operator tries to plan these trips as efficiently as possible on its resource. By planning (and transporting) different trips belonging to different items of freight on the same resource at the same time the operator is actually bundling. The origin and the destination of these bundled trips might be same, but do not need to be (this means there will be intermediate stops). The shipper taking ordering the transport can be the same shipper, but is usually not. The trips can be assigned to the same actor but doesn't have to (this is what Konings is aiming at). The scope and complexity of bundling problems depend on the system at hand and the goals of bundling.

5.2   BUNDLING CONUNDRUM

The previous section left us with a rather broad definition of bundling, but it does not constrain bundling to multi actor problems or other artificial constraints. Both the shipper and the transport operator play a major role in bundling. The shipper's routing enables the bundling of freight, whereas the operator's planning actually bundles the freight. The goal of bundling is to improve the system's utilization and to conserve system's resources. This should improve the cost-quality level of the transportation system or create another advantage over the unbundled situation.

However these advantages must be feasible (there are a lot of constraints to bundling) and there will be some disadvantages (high coordination cost, detours, increased risks and increased complexity). Getting bundling right is an intricate problem, a conundrum, which requires that a trade off is made between its advantageous and disadvantageous results while still complying with the constraints set by bundling and by the system. The criteria used in this tradeoff depend on the system in which bundling is applied. This section gives an overview of the results of bundling and the constraints to bundling.

5.2.1   RESULTS OF BUNDLING

As stated before, bundling is used to create an improvement over the unbundled situation. When applied bundling can have the following advantages:

1. Economies of scale, the actors bundling will have larger transportation flows. The scale of the transportation resources can be adapted to match these transportation flows. Usually the larger a transportation resource gets, the more cost-effective it is. In the case of excess capacity, transport flows can be bundled on existing transportation resources (increasing their utility rate). By using either larger transportation resources or by using the existing resource more effectively, some capacity will eventually be rendered obsolete. The effects of economies of scale are lower overall transportation costs. The more overlap there exists between the networks of the actors involved in bundling, the greater the economies of scale can be. However the network can constrain the size of the transportation resources used.
2. Economies of scope, the actors bundling will have larger network with more origins and destinations to service. This can make the actors more attractive to potential and existing customers because their service provides a better fit with their requirements. The limit lies in the demand for such additional services. To achieve economies of scope the networks of the actors bundling must have some overlap (the networks must interconnect). But it is essentially the lack of overlap that makes economies of scope possible.
3. Frequency, the actors who are bundling have a number of resources which service the same origins and destinations. By bundling the efforts they can service these nodes more often, offering a higher quality of service to their clients. This requires enough resources to be available to the actors.

However there are some disadvantages to bundling, these should be compensated by the advantages. Most disadvantages are caused by the increased complexity; more flows have to be planned on more resource in a larger network. These are the main disadvantages:

1. Detours, the routes of a resource will change because of bundling. In many cases this will mean that resources and freight will travel for a longer distance (and time). This adds to the costs of bundling. Detours will become larger when the number of origins and destinations within a network grows and the size per flow is relatively low (smaller than the capacity of a single resource).
2. Disruptions, the parts of the network where the freight flows are bundled are critical. When these fail the effectiveness of bundling decreases. The resources must be routed in such a way that they are not susceptible to major disruptions in the network; the risk taken must be manageable.
3. Coordination, when multiple actors (usually transport resource operators) are involved in bundling, these actors need coordinate their plans. The coordination of plans can be difficult, due to the size and complexity of plans. This makes coordination expensive.

The results of bundling are first and foremost dependent on the actors involved in bundling, their goals, the network, the resources and the demand for transportation. These set the constraints for the coordinated planning between the actors, the result of bundling are a result of this coordinated planning.

5.2.2   CONSTRAINTS TO BUNDLING

The constraints to bundling determine if bundling is feasible at all. There are two sources of constraints: Technical constraints limit the technical feasibility of bundling and are usually caused by the complexity of the system. Actor constraints are caused by the actors which have to cooperate in order enable bundling.

Technical Constraints

1. Freight flows can only be bundled when they can share a common part of their routes on the same kind of transportation resource and have matching time constraints. When the costs of bundling are larger or equal than separate transportation bundling becomes infeasible.
2. Bundling seems easy, the idea at least. In practice it is not. The resource planning required for bundling means k resources are used to transport n items of freight over m possible routes by o actors. The number of possible solutions is large. When actors bundle their efforts the number of solutions grows exponentially. This means it is hard to find an optimal solution and that coordinating such a solutions puts high demands on the coordination mechanisms used.

Actor Constraints

1. Some bundling types require a reconfiguration of the current network. Nodes can become transshipment nodes; this means that such nodes must have enough capacity to deal with the increase in activity. In extreme cases the network links need to altered, to accommodate the increase in activity. The investments required for such network reconfigurations can be very high, so actors might not be willing to make such investments. Because of social benefits (such as less road traffic) governmental organizations might be willing to invest. This influences the feasibility of bundling.
2. Only when describing the coordination aspects of bundling it became clear that one or more transport operators need to be involved in order to bundle freight flows. The actors need to be able to coordinate their planning processes. This can be inhibited by insufficient communication means, by inaccessible data or even by regulations. Actors must be able to recognize the benefits (and reap them) and they must be willing to coordinate.
3. The benefits of bundling should outweigh the costs of bundling. These benefits should be distributed among the bundling actors in a proportionate fashion. The profit distribution scheme is essential in applying bundling in a multi-actor setting. There are a number of profit sharing schemes based on gaming theory (Engevall, Göthe-Lundgren et al. 2004; Krajewska, Kopfer et al. 2008) which can be applied. The important thing is that profit is distributed fairly among operators.
4. The main reason why actors do not want coordinate in order to achieve bundling lies in the fact that they are usually competing with each other for the same freight, however limited this competition is. Coordination of actions gives actors insight in each other's processes and therefore creates transparency. Others can abuse this information. Either actors must trust the parties with whom they are coordinating, or the transparency must limited. The set of actors can change, the method should be scalable in such a way that it can deal with these changes and that the trust building process does not have to be restarted after each and every change in actor configuration.
5. On the one hand transparency is needed to achieve bundling, on the other hand transparency is what drives actors away from bundling. The double bind caused by transparency needs to be broken. Trust is essential in this case. A few possible solutions exist to escape from this:
   a. Only bundle within an organization, within an organization transparency is the standard. The rewards of bundling can be extended by integrating other transporting actors within the organization.
   b. Create an institution that regulates the access to each other's data and which has the power to punish abuse. In this case the actors remain independent, but are regulated by a third party.
   c. Limit the information exchanged while coordinating, much of the information exchanged doesn't need to be specific and can therefore remain within the bounds of a single organization.
   d. Move the coordination processes away from the operators to a third party. This third party needs to be trusted by the actors involved.

The 'solutions' are not exclusive and can be combined. They represent a number of governance structures which are suitable for application with bundling. The choice for a governance structure depends on the system and the actors involved.

5.3 BUNDLING STEREOTYPES

Bundling takes place in a network. There are a number of ways to organize the bundling network in order to enable bundling; bundling stereotypes. The choice for a bundling network is made for a medium to long term. This is because a change of network would require large investments and would probably mean that the operators in the network have to change the way their resources are routed. The choice for a bundling network can be based on qualitative criteria (Konings 2006) or the much more quantitative method of service network design can be chosen – which constructs a network in such a way that regular services can be offered at minimum costs (Crainic 1999; Wieberneit 2008).

Kreutzberger (2008) defines a number of bundling stereotypes. He starts his classification by the natural distinction between simple and complex bundling networks:

1. In a simple bundling network freight flows have the same origin and destinations nodes within the network. This means the routing remains the same and no network dependencies need to be taken into account while bundling.
2. On the other hand there are complex bundling networks; in this case freight flows have different origin and/or destination nodes. To bundle in these networks transport by multiple resources or detours may be required.

Kreutzberger identifies four distinct complex bundling stereotypes. These describe the simplest case for such a configuration. The stereotypes are described below:

Hub and Spoke Bundling Network

In a hub and spoke bundling network one of the nodes is used as a hub. A hub is usually selected based on the connectivity it provides to other nodes, some of these nodes are hubs in their own right and others are spoke nodes. Cargo is transported from a spoke to the hub. It is transshipped in the hub to another transport which moves it to its spoke destination. All freight flows meet in the hub node and can be bundled in the hub. The distance the freight travels is usually higher in hub and spoke bundling because the freight is first moved to the central node. But by using such a configuration where the hub is always on route, more freight becomes eligible to be transported by the same resource which allows for economies of scale.

Trunk - Collect and Distribute Bundling Network

Using a trunk - collect and distribute bundling network freight flows are bundled in a collection node, these flows remain bundled until they reach the distribution node where they are unbundled and go their own way. In this design collection and distribution nodes are nodes that are not the origin nor the destination of the freight, the path between them, the trunk, is merely shared by the different freight flows, from this perspective such nodes can be seen as hubs. In this configuration bundling takes place on the trunk route.

Trunk - Feeder Bundling Network

In a trunk - feeder bundling network a route covering several nodes is chosen as the trunk. Transport flows that do not have an origin and/or a destination within this trunk are first transported to a node which is part of the trunk. All the freight flows use a part of the trunk and are bundled during the trunk transportation. By picking a trunk which has overlap with (almost) all freight flows, trunk - feeder bundling can be an effective bundling method.

Line Bundling Network

A line bundling network is accomplished by selecting a route (a line) which covers all freight origin and destination nodes. Resources cover the complete line. This means that freight will always be transported, and that bundling advantages can be gained when flows are heading in the same direction. However transports need to cover large distances and the utility varies on different parts of the routes. A line bundling network is the only complex network building block that does not require extra transshipment in an intermediate node.

These bundling stereotypes are often combined, or used together in hierarchical network configurations. This means that one type is used in the main network while others are used in the connecting sub-networks. An example is the intercontinental container transportation bundling network, this is a trunk feeder bundling network, whereas the connecting container hinterland transportation network uses a hub and spoke bundling stereotype.

5.4 BUNDLING IMPLEMENTATION

This section briefly describes how bundling can be implemented. It can only be implemented when the required bundling elements have been designated: the freight, the bundling network, the resources to use and the actors involved. The actors involved must be in agreement over the goal of bundling (which can differ per actor), should trust each other and should have some conflict resolution mechanisms in place.

Implementation Levels

A bundling implementation is a rather broad concept. It depends on the goals of the actors and the requirements set by the context in which bundling is applied. There are three different levels of increasing involvement of a bundling implementation:

1. Enabling: The application allows transport operators to share details about freight which they want to bundle. The transport operators still control their own planning process and use the implementation as a market place at which freight available for bundling is published.
2. Integrating: The application integrates the systems of the transport operators and makes sure information is shared between the systems of the operators. The operators still create their own plans but the information systems are linked and the bundling implementation provides information on freight available for bundling directly to their systems and the takes care of all the communication between the systems of different operators.
3. Controlling: The application bundles freight for the operators and takes control in the planning process. The degree of control the operators have depends on the implementation. This is the most involving level and as this level the commitment of the actors to bundling is substantial.

Once an implementation reaches the third level the implementation has to support the users in planning their resources. For the other two levels planning is done per transport operator, the choice for planning method being their own. Most existing cases do not reach level three. An implementation at level the controlling, third, level uses a planning algorithm for bundling. The used algorithm for bundling depends (again) on the requirements set the by the actors cooperating and the bundling problem at hand.

Bundling Algorithms

A bundling algorithm is used within a level three application in order to create joint plans which are to be executed by the transportation operators involved. There are a number of factors which heavily influence the choice for an algorithm. These and others can be derived from Davidson, Person et al (2007):

1. The bundling elements, the objects involved in bundling and their constellation. They determine the complexity of planning problem. On the one hand the structure of these elements creates complexity by setting bundling constraints and requirements. On the other hand the complexity is determined by the number of elements and their interactions. The more complex a planning problem gets, the more difficult it is to solve it, and as a result bundling becomes more complex.

    As the problem's complexity increases, it becomes harder or even impossible to create an optimal solution for a problem within polynomial time amount of time (in this case the problem is NP-hard). Note that what is considered optimal is determined by the actors doing the bundling: multiple actors can have different views on the term optimal; this can be a problem in itself. However near optimal solutions can be found in reasonable amounts of time – most bundling algorithms find such near optimal solutions using heuristics.
2. The bundling network. The algorithm must be able to deal with the bundling network design and be able to generate plans which use the bundling network as intended. A major in this is the use of transshipment in such a network. Some methods do not take such an option into account and are not directly suitable for use in such networks.
3. The planning intervals. In literature a significant distinction is made between methods that create plans in real time (dynamic methods) and plans that are created in advance (static methods). This distinction is about the time interval in which the method creates a plan which is applicable to the current situation. This is very dependent on the bundling problem at hand, especially on the rate at which the problem changes. It is also dependent on the application of bundling and the actors involved in bundling: when coordination is expensive central planning might occur only once in a while at a predetermined instance, when coordination is cheap planning might occur in real time.

Because transportation is such a fundamental function required by society and because bundling is one of the core abilities of logistics, there is a large body of knowledge on bundling methods. This subsection only provides a small overview of the available methods, not a comprehensive one (however

the most references used refer to such comprehensive overviews). Reflecting on the previously introduced factors a few observations can be made:

1. Algorithms finding optimal solution are only available for relatively small sized problems (finding a solution within feasible time), for all other problems near optimal algorithms are used (Mitrovic-Minic 1998).
2. Bundling problems using a network which requires transshipment are usually solved in two distinct steps (Caramia and Guerriero 2009). The first step is routing of freight to the network. This creates an itinerary for the freight, which contains all the trips the freight will make. These itineraries can be determined in advance using static methods (Wieberneit 2008) or in real time (Marcken 2003) and are usually based on predetermined schedules. The second step is planning the individual trips. The methods for this are discussed in the next observation. Integrated methodologies are hard to find in literature, this can be explained by the fact that solving these problems without the use of decomposition (into separate routing and scheduling problems) further increases the complexity of the problems (and makes the solution of such problems infeasible).
3. Literature mainly focuses on problems without transshipment, in a simple or line bundling network, and focus on the effective use of transportation resources. This is research focusing on the Vehicle Routing Problem (VRP) and the more generic Pickup and Delivery Problem (PDP). Both these problems are NP hard (Savelsbergh and Sol 1995; Laporte 2009), therefore algorithms tackling these problems often use (meta)heuristics to provide near optimal solutions. The following groups of algorithms can be uncovered:
   a. Manual Heuristics. Manual methods require planning by hand by a number of planners (the observations made here are based on fieldwork by the author). Due to bounded rationality planners can only keep track of a limited number of facts. Complexity for planners is reduced by using partitioning the problem into multiple sub-problems (which introduces additional coordination) and by the use of heuristics and guidelines (such as shipping schedules). Only in the simplest cases will manual planning yield optimal results. Manual planning is slow, labor intensive, requires the use of expensive resources and has limits to the size of problem which can be solved by it. However it usually yields satisfycing (Simon 1957) results.
   b. Agent Based (AB). An AB algorithm models the bundling problem domain as a number of agents which try to make a plan (Weerdt 2003; Douma, Schuur et al. 2006). An agent is a software based object which is autonomous, social, pro-active and reactive. Agents come to a plan by interacting with each other. The results of an agent based algorithm can approach those of near optimal solutions. The advantage to these AB algorithms is the relative ease with which a (bundling) problem can be translated to an AB problem.
   c. Operations Research (OR). OR based algorithms use a mathematical description of the bundling problem (usually formulated as a VRP or PDP variant) in which a number of constraints and a cost function are defined. The cost function is minimized without breaking one of these constraints. Because bundling problems are typically nonlinear traditional methods such as the Simplex method cannot be used (Hillier and Lieberman 2005). There are however many types of methods which can deal with these nonlinearities: local search based algorithms, genetic algorithms, dynamic programming approaches, column generating approaches, guided evolution (Savelsbergh and Sol 1995; Berbeglia, Cordeau et al. 2007; Cordeau, Laporte et al. 2008; Laporte 2009).

      An argument against the use of OR in favor of AB methods, is that OR methods increase the transparency between actors (because the data in OR methods is centralized). However Krogt (2007) has shown that AB and OR methods are equally transparent.
   It should be noted that all algorithms will always have a human component to it. An algorithm is always controlled by a planner, who is responsible for a proper planning, in a more or less hands-on way.

5.5 EXISTING CASES FOR BUNDLING

The last section of the chapter looks at the real-world applications of bundling. The purpose is to uncover the bundling networks used, the context in which bundling has been implemented and the effects of bundling on the targeted system. In the previous sections bundling was described from various perspectives: the elements of bundling, the bundling network used, the results of bundling and the implementation of bundling. These perspectives will be used to describe the applications. The usage of the application, the frequency it is used with, and the role of the application within bundling are also discussed.

Code Sharing

In recent years a major consolidation effort has been taking place in the airline world. Airlines operate hub and spoke networks, however the range of these networks wasn't sufficient for the passenger demand for ever increasing destinations (Eggermond 2007). Airlines organized themselves in alliances; these alliances connect the networks of the allied airlines together, thus increasing the scope of the network. By means of code sharing airlines can offers flights which are (partly) operated by allied operators. This has lead to a consolidation of flights over the network, alliances that operated multiple services within for a route, reduced the number of flights; this allows the airlines to consolidate on a number of specific routes. The advantage is that costs are reduced because less services need to be operated, the utilization of services is usually higher and the airline can offer a client better service by providing more destinations. Code sharing works because there are only two airline reservation systems in the world. Code sharing integrates the operations of different airlines. Governance of code sharing is accomplished by the alliances which are central in code sharing.

Intercontinental Container Shipping

Large container shipping carriers operate in alliances of multiple carriers. These alliances operate a number of loops (as explained in chapter 2) in which large container ships of these alliances make a number of stops at deep-sea ports. Although the deep-sea transport takes care of the largest part of the transport, further transport to its final destination is required. In the deep sea-ports the containers due for the ports hinterland are discharged and they put on a follow-up transport (by barge for example). This uses Trunk Collect and Distribute bundling network design. The use of loops allow the carriers to offer regular transport to their customers, by cooperating in alliances carries can increase the scope and scale of their operations. The use of loops integrates the processes of the carriers in the alliance; carriers will transport goods for other alliance members. Loops are commonplace in international container shipping. The alliances take care of governance of their loop system.

Teleroute

Teleroute is an online application in which (small – according to their website) truck transport operators can exchange information on freight and (partially) empty trucks (Teleroute 2009). The system has additional functionality such as route planning. The advantage for truck operators is the low threshold of the system; it enables cooperation between the truck operators and potential clients. However the system creates transparency by openly publishing information of transporters and clients, according to one transporter of 150.000 potential cargo's only 3 are interesting. The system is more an online market place than a bundling application. The system is governed by the Teleroute itself.

Multicasting in Telecommunications Networks

Multi casting is an extreme kind of bundling. Multicast is used in telecommunications networks, to send content in packages to multiple clients (Mieghem 2006). Instead of sending each client a separate package, the package is sent only once, and is only split when the routes to the different clients must split. This can only be achieved due to the fact that exact copies can be made of the content in the package. The package is the resource and the content its freight. The advantage to this is that the bandwidth use of multi-casted content is much lower than its uni-cast counterpart; it depends on the dispersion of the clients and not on the number of clients in the network. The bundling type can be classified as Trunk Collect and Distribute. The governance of such multicast depends on the origin of the content. Multicasting algorithms takes care of the routing and multicasting can be classified as controlling

Company Specific Applications

There are many company specific bundling applications. Most of the literature describing a specific algorithm have done so in cooperation with a logistics provider. There is usually no or limited data available on the companies implementing such algorithms. These algorithms are usually controlling or supporting the operational planning processes.

6   BUNDLING APPLIED TO THE HCSS

This chapter describes how bundling is applied to the HCSS. Over the course of this chapter a proof of concept bundling tool will take shape. The construction of the PoC is the goal of this research (see 1.3.1):

> *The goal of this research is to create a proof of concept bundling tool which shows that bundling of containers on shared barges by VITO ITO's is possible, and that doing so increases the efficiency of transportation (decreases transportation costs) without harming the core values of the parties involved.*

However it was not defined what is meant by a PoC. In essence a PoC is a prototype of the implementation. The PoC is a bare implementation of bundling which only implements the essential elements. The previous chapter identified these elements: the freight, the bundling network, the resources to use, the actors involved, the type of implementation and the algorithm used. Section 6.1 shows that bundling is possible within the context of the HCSS by matching the essential elements to their HCSS counterparts. The next section, 6.2, shows how the HCSS should be organized in order to enable bundling. This section contains the high level (tactical – mid-term) choices made in implementing PoC. Section 6.3 describes the tool used to support planners in creating bundled plans. This section shows how bundling can be achieved on the operational level. This chapter answers the second and third research questions by presenting the design of the PoC and by arguing the design choices made. Concluding remarks and answers to the second and third research questions will be presented in the last section 6.4:

> 2.   *Which bundling stereotypes are suitable for container hinterland shipping?*

And

> 3.   *What bundling algorithms can be used in the implementation of a bundling tool?*

It should be noted that the actual implementation of bundling within the HCSS is beyond the scope of this research. This means that the feasibility of bundling, covered in the next chapter, is largely determined by the tool (especially the algorithm) which is used to create bundled plans (presented in 6.3). The organizational design (see section 6.2) defines the design space available to the design of the algorithm, but is not really used and thus not evaluated. Such an evaluation should be performed in further research, in for example a workshop setting. This is the reason that the focus of the chapter mainly lies with the proposed bundling tool and not as much with the organizational aspects.

This chapter has been based on the results of the previous chapters, literature research, field work at BTCN, personal communications with BCTN's director and chief planner, a brainstorm session with the managers of INITI8, and software prototyping.

6.1   CONCEPTUALIZATION, BUNDLING WITHIN THE HCSS

The section translates the conceptual model of bundling elements to the elements in the HCSS. By making this translation the first step in assessing the feasibility of bundling is taken. This translation is based on the results of chapters 2, 3, 4 and 5.

The first point which needs to be made is that bundling is not a new concept in the HCSS. Barge operators have been bundling transports between the sea-ports and their inland terminals for over twenty years. The main difference between the current and future situations is that the scope of the planning process changes, the scope of the future situation will transcend organizational boundaries and will include multiple inland terminals and the barges operated by these terminals.

The network used is the waterway network connecting the BCTN terminals to each other and the ARA-ports. The network nodes are either intersections or terminals. The network links have practically unlimited capacity and are subject to currents determining travel times, the only links with capacity constraints are lock links. The transporting resources are barges. Barges are like any other resource, except that they have some specific stowage constraints, which can influence the capacity of a barge on a link. The freight to be transported are containers. The origin and destination nodes of a container are always container terminals. A container has a capacity of 1-3 TEU (a 45 Feet container is seen as 3 TEU). Containers have constraints which need to be honored by the transportation process (these have been described in section 2.3).

From an actor perspective BCTN consists of four barge operators: BCT, CTN, CTT and WIT. These operators manage the transport from and to a single inland location with a dedicated fleet of barges using a manual planning approach. The transports flows of these operators will be bundled using the

current capacity available to all operators. In order to create planning synergy the planning of all four operators is done in BCTN's planning office in Nijmegen, with the planners having access to each other's Modality instances. As a result the costs for coordination are rather low. However bundling is only applied on a very limited and incidental basis, because the current set up of the Modality instances create a lot of administrative fuss (for example the bookings need to copied from one system to another) when attempting to bundle freight.

The main goal of BCTN/VITO in bundling is the reduction of costs (i.e. more *effective* transportation). Bundling achieves economies of scale, which enable the reduction of costs. In the case of the HCSS bundling also creates (some) economies of scope and frequency.

The main transportation costs for BCTN are generated by the use of barges. These barges are hired and operated by BCTN. The costs for barges consists of the rent of the barge, its fuel costs (determined by its use), costs for using network resources and port fees. The latter two elements are insignificant compared to the first two so they are not considered. QoS penalties do not occur often, but when they do they are usually closing time related. BCTN tries to prevent these penalties by rejecting transport requests with unfeasible demands and by transporting potentially problematic containers by truck (which is more expensive than transportation by barge). The cost for trucking a container can be used as a proxy for QoS penalties. Because the routing will change it is important to recognize the risks introduced by doing so and the additional costs it will generate.

6.2 ORGANIZATION OF BUNDLING WITHIN THE HCSS

The previous section showed that bundling is possible within the HCSS. This section shows how the HCSS's network and planning processes must adapted in order to enable bundling. By doing so this section answers the second research question.

6.2.1 REQUIREMENTS TO BUNDLING

The requirements to bundling help in selecting a suitable design for the organization of bundling, and will be used in the next chapter for the evaluation. These requirements have been based on the previous chapters and personal communications with all the main stakeholders. The requirements defined in Table 8 should be implemented by both the PoC and an immersive bundling implementation.

Table 8 Requirements for Bundling

	Name	Description
1	Costs	Bundling must reduce the overall transportation costs of the barge operators involved. The costs reduction should be large enough to compensate for the costs of bundling in long run (caused by added coordination, organization of bundling and the implementation of bundling). It is extremely important to note that each of the operators involved should gain from bundling, either directly by using each other's capacity or indirectly by compensation for provided capacity.
2	Robustness	Bundling must protect the robustness of the operation of the barge operators involved, so its results must not be sensitive to small disturbances in execution. Deviation between planning and execution makes promised performance less reliable. This would be a major problem for the other actors affected by bundling (shippers and stevedores); these parties will eventually punish operators for such capricious behavior. When large deviations occur operators are usually forced to allocate additional capacity in order to cope with these effects, which in turn leads to higher costs. Large deviations in plans pose a threat to the trust of actors (all parties, not only barge operators). The robustness of bundled planning should be the same or better as the robustness of unbundled planning.
3	Impact	Bundling must have limited impact on the current way of operations. When bundling requires a large investment, commitment is needed from barge operators and other parties.

6.2.2 BUNDLING STEREOTYPES

Chapter 5 introduces a number of possible bundling stereotypes: Hub and Spoke Bundling, Trunk - Collect and Distribute Bundling, Trunk - Feeder Bundling and Line Bundling. All stereotypes s except line

bundling require transshipment at an intermediate node. Transshipment is problematic for BCTN and VITO members for a number of reasons:

1. A transshipment node (hub) is a terminal, which needs to be able to cope with an increase in containers, barges and trucks. This means more storage space is required and more handling capacity is needed at such a terminal. CTN is the only BCTN terminal which is suitable for this role (location wise). Unfortunately CTN is also the terminal which is extremely limited in terms of storage space. The location of hubs when considering all VITO members is even more difficult, due the geographical dispersion of VITO members.
2. The commitment needed for the creation of a new transshipment terminal is very high. When considering bundling the option for BCTN, they need to make these investments which can be a large strain on the company. In the context of VITO this even more complicated: it is unknown which parties are going to pay for such a terminal.
3. The distances within the HCSS are rather low and the transaction costs caused by a transshipment step are relatively high. These can be higher than the costs for direct transportation, which makes the positive effect of using a bundling method void.

Although the Trunk - Collect and Distribute Bundling, the Hub and Spoke and the Trunk - Feeder Bundling stereotypes are feasible, the impact of the transport network reconfiguration by creating a transshipment terminal is simply too large. This leaves the simplest form of complex bundling, line bundling. The low impact (requirement 3) and potentially easy win in costs appear to make line bundling a good fit for the HCSS.

6.3   A TOOL FOR BUNDLED PLANNING

In the previous section it became clear that line bundling suits the current situation in the HCSS best. This section describes a possible tool which will support the planners in creating bundled plans. The section will describe the need, the requirements, the structure, the verification and the validation of the tool.

6.3.1   EMBEDDING LINE BUNDLING IN THE HCSS PLANNING PROCESS

Currently calls are planned manually. A planner has a batch of transports to plan on a number of resources. This planning problem, a Vehicle Routing Problem (Savelsbergh and Sol 1995; Berbeglia, Cordeau et al. 2007) with more than 3 terminal locations, is NP-hard (Hassin and Rubinstein 2005). In order to deal with such complexity planners use a number of heuristics and their efforts are partitioned per inland terminal. It should be noted that the goal of the current planning is to create plans that are good enough (Moonen 2009), and not to create optimal plans.

By bundling the four separate planning problems (one for each barge operator) become a single planning problem. The scale of the planning problem increases roughly by a factor of four. The planning problem essentially changes from a VRP into its more general form, a PDP, because the constraint that a barge has to return to the same inland terminal is dropped. These changed properties make that an already complex planning problem becomes much harder to solve, while it is perfectly possible to keep using manual planning as the main planning method, this would still require the use of heuristics and partitioning of the planning problem. The limits of man as an effective planner become apparent; as a result the bundled plans created by such a method will not appreciate the chances provided by bundling. In an attempt to reap the fruits of bundling as much as possible, planning has to be supported by an automated tool.

As described in chapter two, the planning process consists of three steps: planning calls for barges, communicating these calls with the terminals called at, and finalizing the calls. The problem with the last two steps is that the communication with most terminals is manual and takes place by means of telephone, fax, internet or e-mail. Some terminals do have services which allow for automatic data exchange by means of EDIFACT or BERMAN messages. Only the actual planning step can be automated to some extent. It is also debatable whether or not a completely autonomous planning step is desirable, autonomous planning would also have to make business critical decisions about resource allocation. Operators (like people in general) have problems with giving up such control. Within VITO a planning step making decisions on resource allocation is completely infeasible as this would take away too much of the operational freedom of individual operators. Also for VITO, because it consists of multiple actors, the planning step must deal with decentralized resource allocation.

After deliberations with BTCN it has been decided that a semi-autonomous planning approach will be taken, in which an automated planning tool will create a pre-planning which planners can use as a guideline in the planning process. The planner remains the central entity in the planning process: he (or she) decides on the barge capacity used, decides how a pre-plan will be used (only parts of the pre-plan

can be used for example), introduces ad-hoc changes to the plans, communicates with terminals and finalizes plans. This makes the planning tool a controlling implementation of bundling (level three, see section 5.4).

6.3.2   REQUIREMENTS FOR A BUNDLING TOOL

The requirements below have been mainly based on the requirements used in the previous section and can be regarded as more specific instances of these requirements (in order to allow for concise referencing the numbering used continues where the requirements in the previous section left off). Note that the requirements defined below are for a full blown implementation of a bundling tool – for the PoC some will be relaxed; this shown by putting the name in Table 8 in italics.

Table 9 Requirements Bundling Tool

Id	Name	Description
4	Validity	The plan created by the bundling tool must be valid within the context of the HCSS. They must honor the constraints defined in section 2.3.
5	Stability	The results of the bundling tool must be stable. The results of the bundling tool for the same data (available barges and container flows) should not result in plans with very different costs. This can happen using (meta)heuristic approaches for a bundling algorithm - getting similar results using such approaches is hard because of the role randomness plays in these algorithms. Very large deviations in plans threaten the trust of barge operators in a bundling tool. Experts and planners indicated that the costs should not differ more than 5%.
6	Accessibility	The plans created by the bundling tool must be easily accessible to and modifiable by the planners. A plan a presented by the bundling tool is a proposition, not a 'fait accompli'. A planner might have a different (not necessarily better) plan in mind, and the planner must have to freedom to implement such change (the planning tool must obey such a change). The planner is responsible for the planning, thus the planner finalizes the plan.
7	Coordination	The bundling tool must limit coordination costs. The more complex a problem is, the more coordination it requires in order to create a feasible joint plan. The bundling tool should provide a mechanism which makes the coordination between barge operators costless or at least limits these costs.
8	Attribution	The bundling tool must enable the attribution of transportation costs/effort made by one operator for the transportation of containers belonging to another operator. This lays the groundwork for a profit sharing scheme, which in turn is the basis for successful cooperation among barge operators. For BCTN the relevance is limited, but within VITO context this is pivotal.
9	Responsiveness	The bundling tool must be responsive. The bundling tool must be able to produce a result of acceptable quality within limited time. When the tool does not deliver results quick enough the added value of the tool will be void, because the situation will change quicker than the tool can plan. The tool must be able to create a new plan for BCTN in the most complex case within 30 minutes. This limit enables the integration of the tool into the BCTN's planning process.
10	Scalability	The bundling tool must be scalable. The entry and exit of barge operators should not cause problems for the use and the results of the bundling tool. The bundling tool should not have any problems integrating with the systems of operators entering and should be easily disconnected from the systems of barge operators exiting. This also means that the bundling tool must be able to cope with the change in complexity; the resulting plans must not suffer in quality (costs and QoS) and planning tool must remain responsive. The fundamental problem with operators entering and exiting is that it changes the elements involved in bundling. The results of bundling are heavily dependent on this and this cannot be prevented by a scalable tool. The tool

|		
		should however aim to make the transition (by entering or exiting) as painless as possible for its users.
11	Flexibility	The bundling tool must be flexible enough to be deployed for barge operators in a different context (for example deployment in a different country). The core of the bundling tool should also be deployable in similar settings outside the world of container transportation by barge (for example container transportation by train). This requirement enables the re-use of the bundling tool other environments. This is a requirement set by INITI8 because it enables them to quickly develop bundling tools for similar problems.
12	Transparency	The bundling tool must limit transparency. The barge operators involved in bundling should not be able to gain enough insight in the business processes of the other operators to such an extent that abuse of such insight in order to get a competitive advantage is attractive.

6.3.3 STRUCTURE OF THE BUNDLING TOOL

This section describes the structure of the bundling tool. It will not go into detail on the implementation but provides a conceptual overview of the tool. The bundling tool consists of a number of components (depicted in Figure 35):

1. Storage, this component provides persistence services for all data required: planning data, evaluation data and reporting data. A SQL Server database is used to implement the storage component.
2. In- & Output (I/O), this component takes care of all the communication services between the Java based components and the storage component. This component is implemented in Java and uses Hibernate (JBoss 2009). In order to allow for flexible I/O operations all the domain classes are Serializable and comply with the JavaBeans standard. In a fully fledged bundling tool the I/O component should also have interfaces with the systems of barge and (potentially) terminal operators.
3. Job, the job component takes care of the execution of planning and evaluation jobs. It monitors for new jobs. It dispatches jobs to the responsible components. It monitors the execution of jobs and it awaits their result. The current implementation is very simple; it just executes the jobs are defined in the storage table and takes care of basic resource management. There is no error handling and no user input is possible.
4. Planning, this component creates plans. The planning component is controlled by the job component and uses the I/O component to load input data from and to write completed plans to the storage component. In order to create plans the component implements a planning algorithm. The planning component uses the same domain classes as the simulation model – this allows for the easy exchange of data between the planning and evaluation components (see section 3.6). Many of the tasks executed in the algorithm can be parallelized (for example the generation of inserts and the calculation of removal scores); only small parts of the code have to be executed in serial. The implementation tries to exploit the parallelizability of the algorithm and the availability of multiple processing cores in modern systems by executing these tasks on multiple threads; especially insert heuristics benefit from this.
5. Evaluation, this component evaluates plans. It is controlled by the job component and uses the I/O component to load plans from and to call evaluations to the storage. The component uses the simulation model described in chapter 3 for the evaluation of plans. This component is a vital part of the PoC as it allows for the evaluation of plans (used in chapters 4 and 7) and is not a required part an implemented bundling tool. It can however be useful for the real-time evaluation of plans.
6. Reporting, this component enables access plans and of plans. The reporting component consists uses a number of queries which extract data from the storage component and produces a number of tables and charts. The queries have been written in SQL. The tables and charts have been created using Tableau, a data visualization tool. The current implementation only allows for the static monitoring of performance indicators and other data required for this research. A future implementation probably will allow both static and dynamic access to much more data.
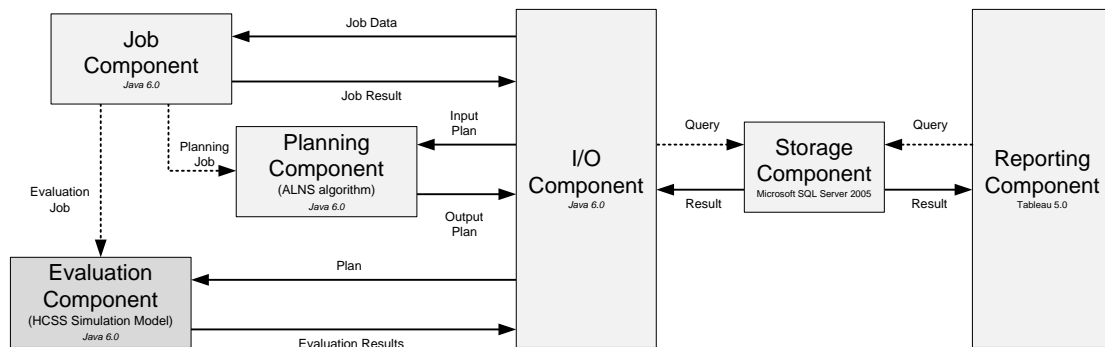
Figure 35 Conceptual overview bundling tool

This is the minimum number of components required for a functioning bundling tool. Not all components have been completely implemented; only the implementations of the planning and evaluation components (the focus of this research) are considered mature. A fully fledged bundling tool should have interfaces with the systems of barge and terminal operators, have an extensive control interface and have extensive reporting component.

6.3.4 BUNDLING ALGORITHM: ADAPTIVE LARGE NEIGHBORHOOD SEARCH

This subsection describes the algorithm used for bundled planning. The algorithm is at the core of the planning component. It is a pivotal part of the tool as it creates the bundled plans; the remaining components provide the required infrastructure which makes the algorithm usable.

6.3.4.1 SELECTION OF THE ADAPTIVE LARGE NEIGHBORHOOD SEARCH ALGORITHM

Transportation by barge can be characterized by its large volumes of containers. This means that a lot of container transports share the same flow (origin and destination). A flow of transports can be seen as a single order. Although this flow can be scattered over multiple transports, it still simplifies the planning problem; for example a window of 2000 transports actually consists of about 86 flows (this means an average of 23 containers per flow) however flows differ greatly in size. This means that the planning problem at hand is actually a specific version of the PDP and can be seen as a PDP with Split Loads (PDPSL)(Nowak 2005). By viewing the problem as a PDPSL, complexity can be somewhat reduced (load splitting in itself is complex) in comparison to the PDP.

The PDPSL is a generalization of the VRP (PDPSL -> PDP -> VRP), these problems are all NP-hard (Savelsbergh and Sol 1995; Hassin and Rubinstein 2005). This means that they are very difficult to solve and that an increase in problem size will result in a non-polynomial increase in effort to solve them. As a result creating exact and truly optimal plans within a reasonable time is infeasible. However, optimality can be approximated by using (meta)heuristics; almost all of the algorithms used to 'solve' such problems are based on this. Exact solvers exists but they can only be applied to relatively small problem instances without having extreme runtimes (Laporte 2009); their merit is mainly in academic environments.

The planning algorithm used must able to solve the PDPSL within the context of the HCSS. The input for such a problem are container transports which need to be transported using barge capacity during a given time window. The solution to such a problem is a static plan which contains a number of planned calls for barges in which the transport requests are fulfilled. A cost function is chosen that minimizes the transportation costs.

There are numerous algorithms which comply with the given requirements: Variable Neighborhood Search (Hemmelmayr, Doerner et al. 2009), Large Neighborhood Search (Shaw 1997) and Memetic Algorithms (Nagata and Bräysy 2008; Velasco, Castagliola et al. 2009) for example. The best results of these algorithms are within 1% of an optimal solution (Laporte 2009). The algorithm selected must also be flexible enough to deal with the unique constraints set the HCSS. The Adaptive Large Neighborhood Search (ALNS) algorithm has been selected. The algorithm has been chosen based on its flexible structure; it will allow for modifications needed in order to deal with split loads.

6.3.4.2 ADAPTIVE LARGE NEIGHBORHOOD SEARCH APPLIED TO THE HCSS

The following subsections contain the description of the ALNS algorithm applied to the HCSS. The contents of these subsections is largely based on the articles by Røpke and Pisinger (2006; 2007) on the subject.
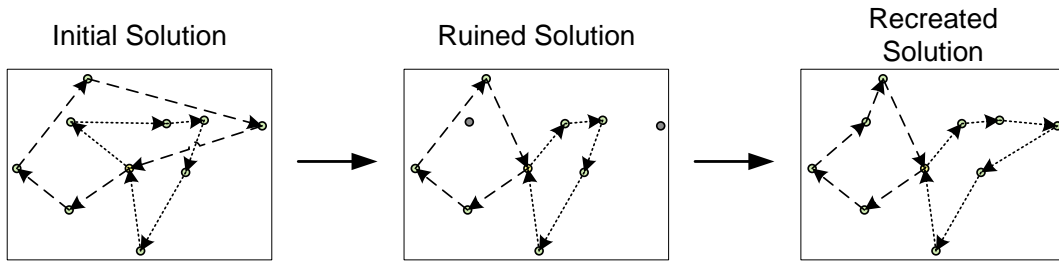
Figure 36 The application of the ruin-and-recreate strategy to a simple routing problem

The ALNS algorithm is an extension of the Large Neighborhood Search (LNS) algorithm (Shaw 1997; 2006; 2007). ALNS and LNS belong to family of local search algorithms. The ALNS algorithm iteratively improves the current solution by exploring related (neighboring) solutions (Ropke and Pissinger 2005; Pisinger and Røpke 2007). It does this by applying a ruin-and-recreate strategy (Schrimpf, Schneider et al. 2000): first the algorithm ruins a large part of the current solution and then it tries to (re)create the best possible solution from the ruined one – the recreated solution is possibly an improvement over the current solution. An example of the application of the ruin-and-recreate strategy has been depicted in Figure 36 – in this case the strategy leads to an improvement.

The algorithm ruins the current solution by means of the removal of a number of requests from the current solution by a removal heuristic. The removal heuristic places these in a request bank which holds all unplanned requests. Resource capacity is freed by removing requests; this relaxes the solution. The algorithm tries to (re)create the best possible solution by using an insert heuristic which tries to insert all requests in the request bank at their lowest cost point in the relaxed solution. The insert heuristics will not stop until all options for insertion are exhausted or no more requests are left in the request bank; this means that after running the insert heuristic some of the request might still be unassigned.

The goal of the algorithm is to minimize the costs for a solution. Usually the costs are expressed in the form of distance, transportation capacity used or in a combination of both. The ALNS algorithm for use in the HCSS will use the costs function as defined in section 3.5.2 which represents the total costs for transportation by barge and alternative modalities.

Instead of using one heuristic for removing and for inserting, the ALNS uses several heuristics for both removal and insertion. The use of several heuristics creates the opportunity to use different strategies to target different parts of the problem. For each iteration one removal and one insert heuristic is selected. The selection of these heuristics is based on their previous performance. The more a heuristic contributes in creating better solutions, the better it will perform, the more likely it is to be selected in the future. The algorithm tries to use the right tool at the right moment by adapting the use of heuristics to their performance. This is why it is called Adaptive LNS.

6.3.4.3 REMOVAL HEURISTICS

The ruin stage is performed by one of the six possible removal heuristics. Røpke and Pisinger (2005) use heuristics that operate at the transport request level. The problem with using such heuristics in the case of the HCSS is that the removal of a single transport request has little effect on the overall solution. This is caused by the fact that requests are treated in batches, the calls, and the removal of a single transport is unlikely to lead to the removal of the entire call. In an effort to create more effective removal heuristics the ideas of three of the removal heuristics described by Røpke and Pisinger (2005) have been applied at the call level. There is one minor drawback in removing calls from a solution; the resulting number of requests removed as a consequence of the call removal is harder to control. The removal of calls will continue untill at least the targetted number of requests has been removed.

Random Removal

This heuristic removes q requests by either removing transports or calls. The goal of this operator is to diversify the search. The random removal of transports truly achieves this goal. However the removal of calls is a much more coarse mechanism for removal; relatively large amounts of similar requests (sharing the same load or discharge terminals) are removed.

The operator works by removing a random item from the calls or transport requests in the solution (step 1). The algorithm repeats step one until at least the targeted number of requests has been

removed from the solution. It goes without saying that all the removed requests are moved to the request bank upon removal.

Related Removal

Related removal removes calls and transport requests which are somehow related – requests in related calls are also related (they share a similar terminal location). Because of the similarities it is expected that the removed requests are relatively easy to re-insert, which should lead to better solutions.

The heuristics initializes by randomly removing an item from the calls or transport requests in the solution and by storing it in a reference set (step 1). The next step is to get a reference item from the reference set by randomly selecting one of the items in the set (step 2). For the items left in the solution the relatedness between them and the reference item is calculated (step 3). In order to prevent myopic behavior (i.e. shortsightedness: the heuristic only optimizes and fails to produce diverse and potentially better solutions) the algorithm does not nessecarily removes the most related item, but uses a random measure to remove the k-nearest related item (step 4). The item removed is added to the reference set (step 5). The heuristic repeats steps 2-5 until at least q requests have been removed from the solution.

The relatedness measure for transport requests is a simplification of the one suggested by Røpke and Pisinger. It will be based solely on the distances between the transports respective origins and destinations. The relatedness measure of calls is somewhat different. In order to calculate the this type of relatedness two key properties of these calls have been used: the distance between the called terminals and the difference between the calling time windows.

Worst Removal

Worst removal tries to remove calls and transport requests that are very expensive. These items are typically placed in the wrong position in the solution; this means that they can be reinserted at a different position at lower costs. Removing these items creates opportunities for cost-effective reinsertion.

Worst removal works by the use of a special cost function which can determine the effect of the removal of an item (call or transport request); i.e. the change in costs. The heuristic starts by calculating the cost change for each call or transport request in the solution (step 1). In order to prevent myopic behavior the k-worst item is removed using the same selection mechanism as the related removal heuristic (step 2). The heuristic repeats steps 1-2 until at least the targeted number of requests has been removed from the solution.

The special cost function is an approximation of the difference in costs between a situation in which the item is a part of the solution and the situation in which it is removed from the solution altogether (the items request are not placed on the request bank). The costs for call removal are approximated by taking the costs for transportation to and from the call and the costs for call handling into account. Absolute call costs are not really comparable because there are differences between the number of moves performed during calls. In order to make them comparable the costs are normalized by the number of moves made during the call. When a call is removed from the solution the requests within the call are also removed from their opposite calls. This not taken into account in the current cost approximation; costs will even change more.

The costs for request removal are approximated by using the combination of costs of the load and discharge calls. The current approximation does not take the actual transportation costs into account, it only shows the effect removal will have on the solution costs. Experiments using the transportation costs of a request did not yield better results and the performance of the algorithm deteriorated because of the more elaborate calculations required.

6.3.4.4   INSERT HEURISTICS

The insert heuristics used with the algorithm are the same one as suggested by Røpke and Pisinger (2005). The insert heuristics only target transports, when a transport gets inserted at a location at which no suitable call exists, the call and the two required hops are inserted at that location. There are two different heuristics used for insertion: greedy insert and k-regret insert. The structure of these heuristics is exactly the same, except for the strategy used in selecting the best request to insert.

Basic Insert

The basic insertion heuristic used is a parallel heuristic, this means that all the routes in the solution are built at the same time (Potvin and Rousseau 1993). The algorithm generates an insert per route for each transport request in the request bank (step 1). An insert is the combination of a load and discharge call at the lowest cost at which the request can be inserted into a route. Neither the load nor the discharge call has to exist before the insert is generated; these 'virtual' calls are materialized when the insert is applied to the solution. The way the inserts are generated has been based on Campbell and Savelsbergh (2004) and Potvin and Rousseau (1993). In order to prevent myopic insert selection some randomness is applied to costs of the insert. The 'best' insert is selected using the selection strategy employed by the heuristic (step 2). The 'best' insert is executed (step 3); adding the requests to the calls defined in the insert and removing the request from the request bank. The insertion of requests, steps 1 to 3, continues until no more requests are left in the request bank or when no more feasible inserts can be generated for the unplanned requests.

The algorithm generates inserts for n request in each of m routes, leading to $n * m$ inserts. The generation of an insert for a request in a route has minimum complexity of $O(n)$. This means that step 1 of the algorithm has a minimum complexity of $O(n^3)$ (Campbell and Savelsbergh 2004); the creation of these inserts can be extremely expensive. However the application of an insert only changes one route; this means that the assumptions for this route become invalid and that only the inserts for this route have to re-generated. During the first iteration the first step of the algorithm has to generate $n * m$ inserts. Only $n - k$ (k being the number of requests already inserted) inserts need to be re-generated during following iterations. This is a significant speed-up of the algorithm.

Greedy Insert

This insertion strategy selects the insert with the lowest costs. The problem with this strategy is that the hard (expensive) requests are deferred until later iterations, a moment at which routes suitable for transportation might be full already – so the requests cannot be placed.

K-Regret Insert

The k-regret insertion strategy tries to deal with hard requests. Instead of only looking at the best insert for a request, the strategy looks at the difference between the best insert and the k-best insert for a request (Potvin and Rousseau 1993; Røpke and Pisinger 2006). The bigger the difference (regret), the higher the costs for inserting the request later on will be (i.e. we will regret not inserting the request as this point). The strategy selects the best insert for the request with the highest regret value.

K-Regret based insert is actually not one heuristic, but a family of heuristics all with a different k value. For these heuristics to be meaningful the value of k should be between 2 and the number of routes in the solution m. Only a limited number k-regret insert heuristics are used at the same time. Røpke and Pisinger (2005) argue that different k values yield different results, high k-values for instance detect quite early on that the possibilities for inserting a request become limited.

Insert Heuristics in the HCSS

The proposed insert heuristics do not explicitly use the structure of the HCSS. The inserts can take advantage of a number of things. Most requests come in batches, these batches share the same properties, this makes the requests in the batch exchangeable; only one insert per route needs to be created per batch. Currently this has been implemented using a caching mechanism – which identifies the requests of the same group and assigns the same request to it. This significantly speeded up the insertion algorithm.

The batch properties of the inserts can also be used to defer recalculation of route data. Batches share the same insert, so all items in the batch have the same insertion costs. When the best insert is part of such a batch, the complete batch can be inserted at once. This has not been done in the current implementation of the algorithm, but should be investigated because it can dramatically reduce the number of computations required by the insertion algorithms.

6.3.4.5   HEURISTIC SELECTION AND ADAPTATION

The algorithm uses a number of competing heuristics for both the ruin and recreate steps. Each heuristic has a weight associated with it; the higher the weight, the more probable it is that the heuristic will be selected. The mechanism used to select a heuristic can be best described as roulette wheel selection (Goldberg 1989).

The weights could be fixed but this would require manual tuning which would, given the amount of operators, not be an easy undertaking. The adaptive mechanism keeps track of the performance of the heuristic during a segment of the run. The performance of the heuristics is represented by a score. The mechanism adds points to a score when the solution is created using the heuristic. The heuristics are only rewarded for suggesting solutions that diversify search. After n iterations the mechanism updates the weight.

6.3.4.6   SIMULATED ANNEALING

The ALNS algorithm is mainly targeted at finding of new solutions. It does not determine how the search should be conducted; a local search framework is used for this. There are numerous local search frameworks usable e.g.: Hill Climbing, Tabu Search, Simulated Annealing, Threshold Acceptance, Guided Local Search and Great Deluge. From these frameworks Simulated Annealing (SA) has selected because of its ability to deal with local optima, the previous successful use of the SA framework in combination with the ALNS algorithm and the ease of implementation.

SA prevents myopic behavior and escapes from local optima because it will not only accept improvements as the current solution (as for example Hill Climbing does), but also allows for regressing solutions (i.e. a downhill move) (Press, Teukolsky et al. 2002; Weise 2009; Wikipedia 2009 Simulated Annealing). Regressing solutions will only be accepted with a probability which is dependent on the cost difference between the suggested and current solutions and a 'temperature' parameter; this is based on the Boltzmann probability distribution (Press, Teukolsky et al. 2002). The higher the temperature parameter, the higher the chance that a regressing solution is accepted as the current solution.

The ALNS algorithm will run for a predetermined number of iterations (25000). During the run the temperature is gradually decreased. This is done using a basic cooling schedule in which the temperature is decreased after every ruin-and-recreate step. When the algorithm is unsuccessful at finding an improved best solution for a number of iterations (5000), the current solution is replaced by the best solution so far; the algorithm is restarted.

The temperature of the SA framework is related to the problem at hand. The temperature is initialized by using the transport costs (not the overall costs – these can be extreme due to a large number of unplanned requests) of the initial solution. The temperature is chosen in such a way that a solution that is a factor (5%) worse is accepted as the current solution with a probability of 50%.

6.3.4.7   INITIAL SOLUTION

The initial solution is the starting point of the search, and has a large influence the outcome of the search (Bräsy and Gendreau 2005). In day to day operations the input for the algorithm would be the plan of the day before adapted to the current situation. Parts of the initial plan should not change (appointments with terminals, special requests made, etc…) so they are fixed and cannot be altered by the algorithm. As a result of a partially fixed plan the degrees of freedom are limited, this means that the algorithm can run faster because of the smaller search space. During the PoC phases the bundling algorithm is not used in such a context; the initial plan is empty.

The initial solution passed to the algorithm will be augmented by attempting to insert all unplanned requests into the routes within the solution. For this a greedy insert heuristic is used.

6.3.5   VERIFICATION & VALIDATION OF THE BUNDLING TOOL

This section describes the verification and validation of bundling tool. The verification and validation process has two goals: the discovery of defects in the tool and to check if the tool is useful and usable in a planning environment (Sommerville 2006 Chapter 22). This section is organized in the same way as section 3.7: syntax, semantics and pragmatics.

Syntax

The syntax is mainly enforced by the platforms used: Java and the Microsoft SQL Server. These platforms check for coding errors and provide meaningful traces when errors occur. The environments add to this by providing syntax coloring and correcting facilities. There have been added some explicit checks to the code in order to prevent illegal object states (for example Not a Number values or deferred null values). The 'advantage' of syntactical errors is the relative ease with which they can be uncovered, the result of such an error being that some functionality (in some cases the entire application) does not work; the tool is syntactically correct.

Semantics

In order to the check if the tool is semantically correct a number of tests have been performed. Unit tests (Wikipedia 2009 Unit Testing) have been performed on parts of the code in order to guarantee the proper behavior of this code. A peer-review of the code yielded only minor improvements.

One of the main methods for checking the semantics is the inspection of the results of the tool. There are a number of results the tool delivers: data in the databases, visual reports and logging files. A number of errors were uncovered by inspecting the results:

1. Using visualization it was uncovered that between some calls no hop was performed. This was the result of a coding error involving invalid storage of calls. This has been fixed and all calls behaved correctly.
2. Errors in the capacity code have been uncovered when the plans were visualized and when the logs were inspected. A barge would sometimes use more capacity than its maximum capacity. The cause was an error in checking the capacity.
3. The algorithm would plan all calls on the cheapest barge. This was caused by the fact that the implementation 'forgot' to implement a proper time window. After the time window was added the planning algorithm behaved properly.

An expert reviewed the plans created by the tool and stated that the plans were valid, but that the routing of barges through the network was sometimes a bit strange; barges usually do not visit two seaports and one inland terminal in a sequence. The expert could not tell if it was wrong, it just seemed strange and more expensive to do.

The HCSS simulation model was an unexpected source of verification and validation. The model implemented some of the constraints in a more elaborate and more precise way; the planning tool used assumptions to cover the same constraints. When the simulation model could not successfully execute a plan, either the model or the plan was incorrect. This cross validated both the model as the algorithm (and its results). An example of this has been described in section 3.7.

Pragmatics

In order to check the pragmatics of the tool the requirements set in subsection 6.3.2 are revisited. The goal is to show which requirements have been taken into account during the design of the bundling tool and which requirements still need attention. Table 10 shows how each requirement has been implemented and what the score of the tool is for the given requirement ('-' = bad, '+/-' = needs some work, '0' = neutral, '+' = good and '?' = unknown). It shows that the tool can be used in its role of PoC. For a complete implementation a few extra steps need to be taken as discussed in section 6.3.3 and some of the requirements need additional attention.

Table 10 Implementation of Bundling Tool Requirements

	Name	*Implementation*	*Score*
4	Validity	The validity of the plans is guaranteed by the implementation of the domain classes and planning algorithm.	Good
5	Stability	No measures have been taken – The plans are always constructed in the same way. This is not a guarantee for stability, but analysis has shown that results are within a margin of 5%. See section 7.1 for a further elaboration.	Good
6	Accessibility	Plans can only be read, not altered. This means that accessibility is very limited. This is mainly caused by a lack of proper user interface. The backing infrastructure has been implemented in such a way that accessibility can be added at a later stage.	**Needs Work**
7	Coordination	The current tool has no support for limiting coordination costs. During the implementation of I/O component a number of measures have been taken that allow for the tool to easily coordinate with other platforms.	**Needs Work**
8	Attribution	The attribution of costs is implemented. The components are able to calculate the costs of a transport and contain information on the	Good

9	Responsiveness	The algorithm and the components that have been designed are responsive. This starts with the choice of platforms used for the implementation and ends with optimization of different components, for example parallelization of the algorithm and the use of a heavily optimized library for persistence. Currently the tool is responsive enough: it takes a maximum of 21 minutes to create plans for the HCSS under the heaviest load (2200 transports). A further quantitative evaluation is presented in section 7.1, as a part of the evaluation of the impact of bundling tool on the planning process.	Good
10	Scalability	The algorithm and the components that have been designed can be scaled very well. For example the algorithm uses parallelization in order to scale well under increasing problem complexity. This shows in the evaluation in 7.1, where the run times of the tool almost scale in a linear way.	Good
11	Flexibility	The algorithm chosen is extremely flexible and can be adapted to different problems by adapting the heuristics used. The components can be easily reused in similar cases, for more generic cases more work is required.	Good
12	Transparency	No measures have been taken. This can implemented using a proper security infrastructure.	**Bad**

6.4   CONCLUSION

Over the course of this chapter bundling has been applied to the HCSS. A bundling organizational design and an algorithm capable of creating bundled plans have been selected and implemented in a bundling PoC. By doing so research questions 2 and 3 have been answered.

> *2.   Which bundling stereotypes are suitable for container hinterland shipping?*

The actors selected for bundling are the four separate barge operators internal to BCTN: BCT, CTN, CTT and WIT. Because of this, issues such as coordination, transparency, resource allocation and cost attribution are of less concern than in a situation in which truly external operators are involved (such as in the case of VITO). The goal of the research is evaluate the effectiveness of bundling applied to the HCSS, not to create a fully fledged bundling implementation (in which other barge operators can play a part). All regular container transports between the inland terminals and the seaside terminals are to be bundled on the joint capacity of the four selected operators.

There are four bundling network stereotypes: Line, Hub-and-Spoke, Trunk-Line Feeder and Trunk-Line collect and distribute. Line bundling is the only stereotype that does not require a transshipment terminal. Transshipment terminals are problematic because of a number of reasons: CTN is extremely limited in terms of space and is the only terminal suited for the role of transshipment terminal, the investments and commitment for creating a new terminal are high and transshipment causes extra transportation costs. The leaves line bundling as the only feasible bundling network stereotype. The low impact and potentially easy win in costs appear to make line bundling a good fit for the HCSS.

> *3.   What bundling algorithms can be used in the implementation of a bundling tool?*

The planning problem the algorithm needs to solve is a version of the pickup and delivery problem with split loads (PDPSL). Groups of containers with similar properties are seen as separate load. These loads need to be picked up at one terminal and transported to another terminal without violating constraints. These loads can be transported using one or more barges.

The pickup and delivery problem with split loads is NP-hard. This means that exact algorithms can be ruled out as feasible algorithms. There are numerous highly competitive algorithms which can provide near-optimal solutions to the pickup and delivery problem with split loads. The Adaptive Large

Neighborhood Search has been selected because of its flexible nature. This algorithm can be easily adapted to the unique challenges of hinterland container shipping.

This chapter showed how a bundling implementation can be shaped. This resulted in a bundling proof of concept capable of creating bundled plans for the HCSS. The next chapter will use this PoC concept in order to evaluate the effectiveness of bundling applied to the HCSS.

7 EVALUATION OF BUNDLING WITHIN THE HCSS

This chapter evaluates the situation in which bundled operator planning (BOP) is applied to the hinterland container shipping system (HCSS). This answers the last research question:

> 4. *What are the costs of hinterland container shipping using bundled operator planning?*

The bundling organizational design and PoC will be applied to the experiments defined in chapter 4. The effect of bundling is evaluated by comparing the costs and effects of Bundled Operator Planning (BOP) to the costs and effects of Single Operator Planning (SOP) (see section 4.4).

The first section of this chapter shows if the bundling PoC is compatible with the planning process of a barge operator. The second section evaluates the bundled plans by comparing them to their unbundled counterparts. In section three it is determined under what circumstances bundling is feasible for barge operators. The final section concludes this chapter and answers the 4th research question.

In order to create bundled plans, resource planning has been done by the PoC. The PoC always started with the same set of thirteen barges on the same locations and was free in selecting which barges to use. This means the PoC was relatively unconstrained in planning resources, whereas in real world experiments the planning of resources will be much more constrained, because of barge operator preferences. This must be kept in mind while evaluating the experiments.

7.1 EVALUATION OF THE BUNDLING TOOL

The first section of this chapter evaluates the bundling tool and algorithm. The operation of the tool determines how much the day to day planning operation of a barge operator would be influenced by the usage of such a tool.

Stability

The results of the bundling tool for the same data (available barges and container flows) should not result in very different plans with very different costs. The results must be stable; this has been stipulated in requirement 5.
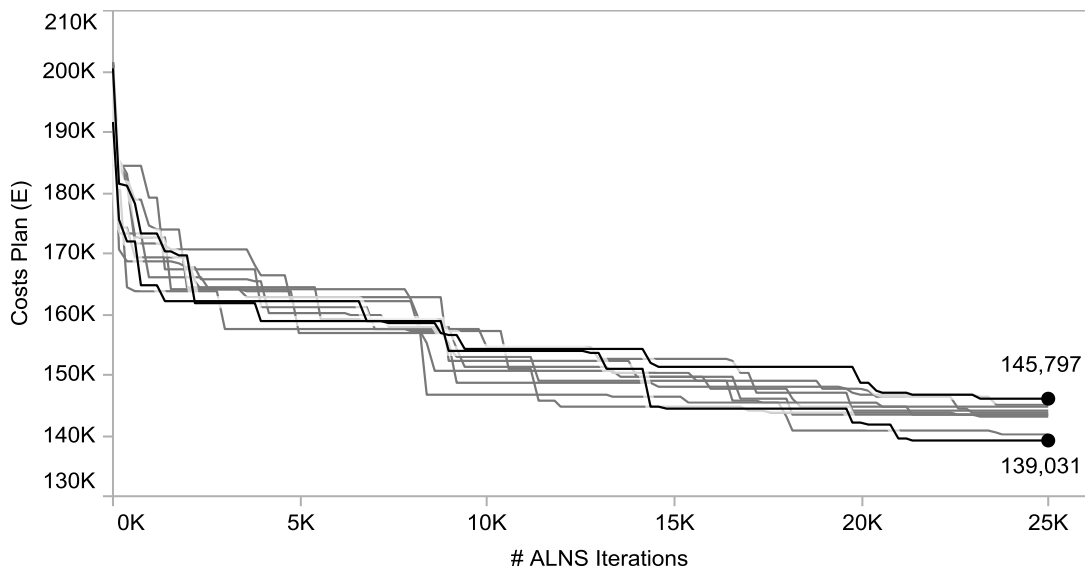
Figure 37 Cost of plans for Experiment E[115890] as the algorithm progresses

The stability has been tested by running the bundling algorithm, 10 ten times on experiment E[115890], thus creating 10 different plans for the same problem. Figure 37 shows the execution of these 10 runs by depicting the development of the overall costs as the algorithm is progressing. What is apparent is that results of the algorithm are not the same per run, this is caused by the randomness used in the algorithm. The results differ as much as 5%, which is just within the limit set.

The results show that there a different feasible solutions for the same planning problem, some better than others. It might be interesting to compare the robustness of the different plans. The stability has been tested on an experiment in the most complex category (2200 container transports to plan); it

would be interesting to see if the algorithm would produce more stable plans in less complex planning instances; the hypothesis is that it does. Both these subjects can be addressed in future research.

Responsiveness & Scalability

The tool must deliver plans within the required planning interval for the tool to be usable by the planners within BCTN; the tool must be responsive. It must remain responsive as additional operators get involved in bundling cargo; the tool must be scalable.
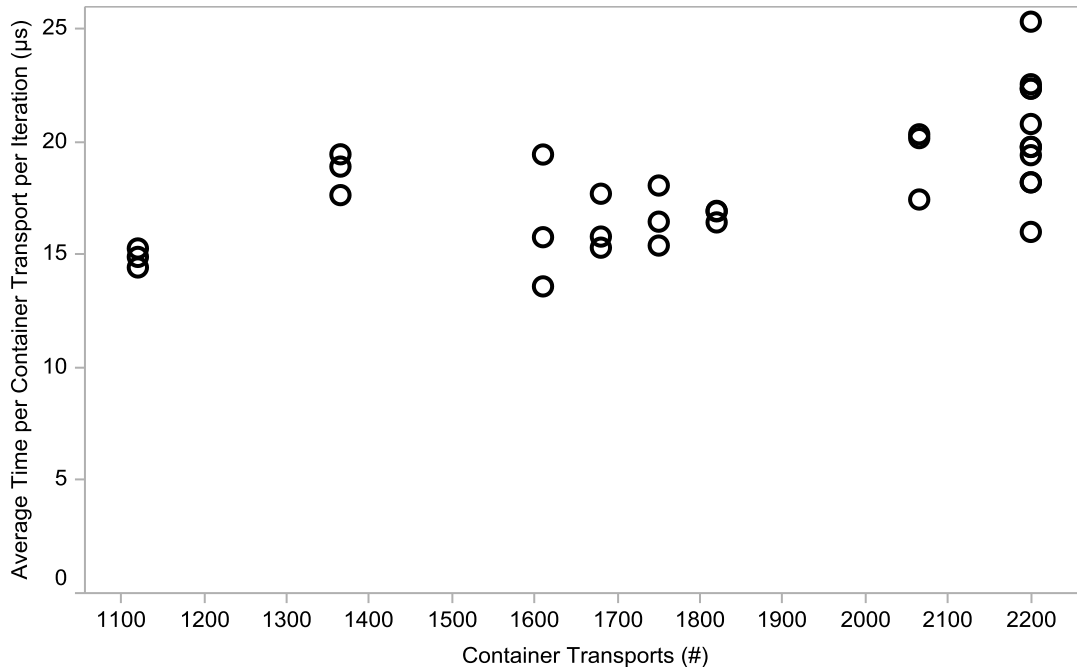
Figure 38 Average time per transport per ALNS iteration (µs – micro second) per Experiment

Figure 38 shows that the average time per container per ALNS iteration is somewhat dependent on the number of transports; as the transports increase the time per container increases as well. The large deviations in performance for the same sized experiments can be explained by a number of factors:

1. Differences in the availability of processor capacity for planning. Measures have been taken to prevent exactly this: all the plans have been made on the same machine and using the same configuration. In spite of this effort this was not tested under the right circumstances. This makes it hard to draw conclusions on scalability.
2. Differences in the planning complexity. These differences could be caused by the neighborhood of the solution the algorithm uses for its current solution. Insertion can take a long time for some neighborhoods – especially when a number of routes are very similar.

That being said, it still thought that the tool scales rather well: the number of containers doubles, whereas the average time per container increases by only 33%. The duration per iteration increases because both the ruin and the recreate heuristics have to perform extra iterations in order to get the required result.

The reason for the good scalability can be found in the ruin heuristics which are used; these heuristics remove a maximum 125 transports (or a number of calls which is equivalent to this) from the solution. The maximum of 125 transports has been determined by experimentation; it allows for proper exploration. The maximum should be changed as the scale of the problem changes; as a result the responsiveness will deteriorate. About 80% of the time is spent in the recreate heuristics which have been designed for parallelization; meaning that responsiveness can be easily improved by adding extra processing cores.

Currently the most intricate plans are constructed in 23 minutes. This is well within the set goal of 30 minutes. Batched inserts and other improvements can further reduce the processing time.

Flexibility

The tool must deployable in environments with different planning problems. The adaptive mechanism and the plug-in style with which heuristics can be added should accomplish this. This section illustrates the flexibility of the algorithm by showing how it automatically adapts the weights of heuristics in order to stimulate the use of the more effective heuristics.

Figure 39 shows the weight evolution of the different ruin heuristics while running experiment E[96495]. The weights are adapted according to the performance of the ruin heuristic used. As run progresses the costs converge and it gets harder to find new (optimal) solutions; a succesfull application of a heuristic becomes harder and the weights of every heuristic decrease. The adaptation of the algorithm is especially shown by the Worst Removal Heuristic used, during the first ~21000 iterations transport removal is dominant whereas during the last 4000 iterations the call removal takes over; the algorithm adapts to a different and probably more constrained solution space in which larger moves work better.
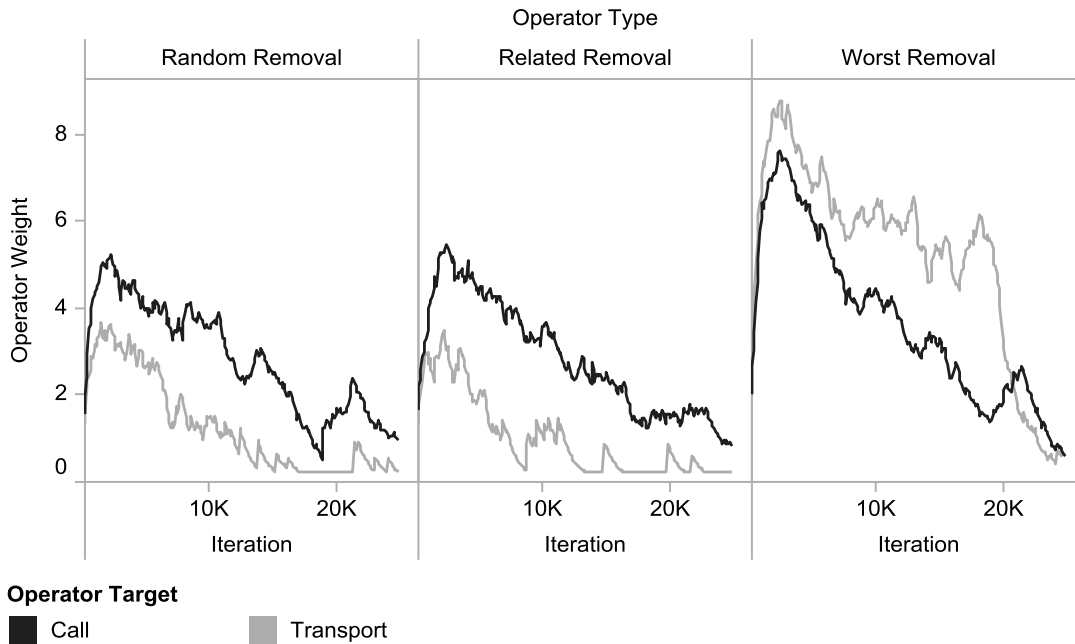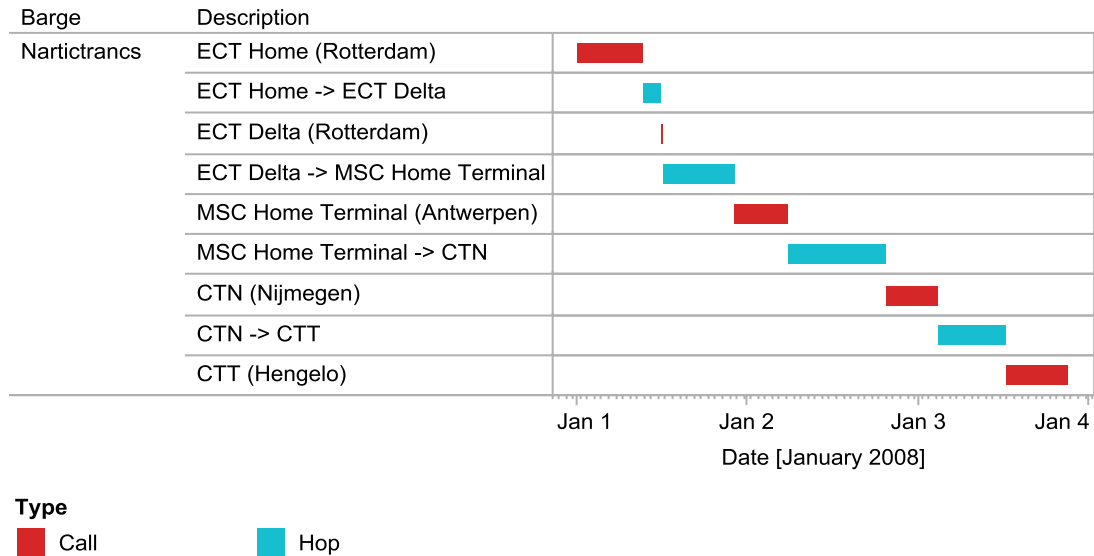

Figure 39 Operator weight per Operator Type

On a final note on the use removal heuristics: Figure 39 shows that worse removal is somewhat dominant over related and random removal. This is surprising since it is a deviation from the results that Røpke and Pisinger (2005) reported: these clearly show that related and random removal heuristics are dominant. A related, not so surprising result is the higher effectiveness on call removal in all but worse removal. This can be explained by the fact that more of the solution is ruined by call removal and that the algorithm can make larger, more effective, moves through a heavily constrained neighborhood. These results are the same for all other experiments.

Recognition, accepting bundled plans

This final step in the evaluation of the bundling tool is not explicitly mentioned in the requirements for the bundling tool. While making the requirements it was assumed that a valid plan would be automatically accepted by planners. However, during the evaluation of plans an expert raised some questions about some of the plans; in his opinion these plans were rather strange looking and would not be created under normal circumstances. Figure 40 shows an example of such a plan, the barge Nartictrans visits the ports of Rotterdam, Antwerp, Nijmegen and Hengelo, especially a visit to two sea ports seems a bit much. The expert noted that he was less inclined to accept such a plan because of its uncommon structure, he is more inclined to accept more visually pleasing plans – i.e. the simpler the better. This preference for pleasing plans has also been observed in previous research (e.g. Lu and Dessouky 2006). Further development of the algorithm could add constraints to or use different evaluation criteria for the create heuristics in order to create more visually pleasing plans.

Barge	Description	
Nartictrancs	ECT Home (Rotterdam)	
	ECT Home -> ECT Delta	
	ECT Delta (Rotterdam)	
	ECT Delta -> MSC Home Terminal	
	MSC Home Terminal (Antwerpen)	
	MSC Home Terminal -> CTN	
	CTN (Nijmegen)	
	CTN -> CTT	
	CTT (Hengelo)	

Jan 1   Jan 2   Jan 3   Jan 4

Date [January 2008]

Type

Call   Hop

Figure 40 Unusual plan for Nartictrans E[90300]

The current bundling application is a relatively unconstrained. The tool only honors critical 'technical' planning constraints (as described in 2.3), more 'human' constraints have not been taken into account. The current tool shows the maximal effect of bundling, adding more constraints may reduce the effectiveness of bundling (adding constraints reduces the solution space, as a result the performance of the algorithm can actually increase). The system can perform with any set of constraints, so deciding on the constraints is a problem to be solved with the involved actors.

7.2   EVALUATION OF BUNDLED OPERATOR PLANNING

This section evaluates the costs and effects of bundling on the performance of the HCSS. This section uses the results of section 4.4 as a yard stick for evaluating the effects. This section uses the same structure as section 4.4 in order to maintain readability. However the goal of this section is not be a carbon copy of section 4.4 but to illustrate how Bundled Operator Planning differs from Single Operator Planning.

7.2.1   ANALYSIS OF BUNDLED OPERATOR PLANNING

The plans are the actual output of the planning process. This section performs a static analysis on the plans and shows how resources are allocated using BOP, and compares this to SOP.
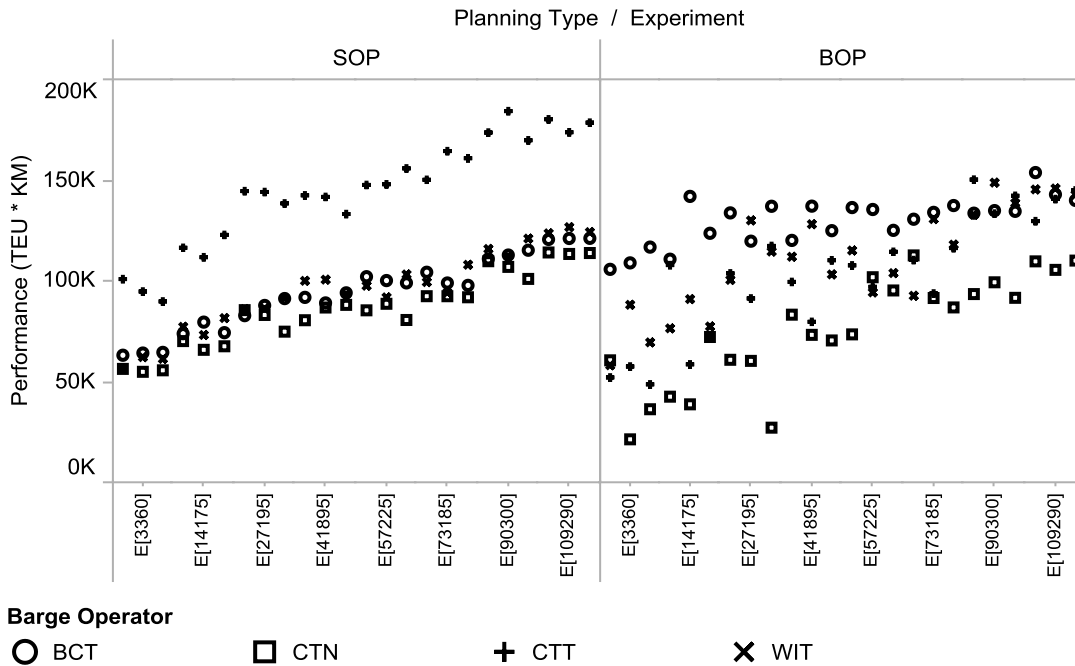
Barge Operator Performance

Bundling does not change the overall transport performance delivered on the short term; on the long term the performance could change due to costs effects as a result of bundling. It does however change the performance delivered per barge operator; some operators will transport less and some others will compensate for this.

Figure 41 shows the performance delivered per barge operator per experiment per planning approach (the experiments have been sorted form small to large). The difference between the SOP and BOP approaches shows the effect of bundling on the allocation of barge operator resources; the more performance delivered, the more capacity needed. The SOP approach shows a gradual increase in the performance delivered by an operator as the demand for transportation increases. The BOP approach draws a very different picture:

1.   The delivery of performance is more scattered than in the SOP variant. The total delivered transport performance for both planning approaches is the same. This means that there are barge operators executing transports for other barge operators; work is assigned differently among operators.
2.   BCT delivers much more performance than in the SOP variant. In all but the smallest experiments the performance delivered is around 135 K TEU*KM; which is probably the maximum achievable performance for BCT. BCT gets a much higher work load than other terminals because the transportation resources used by BCT are cheaper per unit (not per container) and they can easily reach all BCTN terminals.

3. The performance delivered by CTT is much lower than in the SOP variant. This can be explained by the fact that CTT's delivered performance was partially based on the use of truck as an alternative mode of transportation. BOP uses the capacity of other barge operators in order to deliver the performance requested by CTT's clients.

4. The choice for performance delivery by CTT, CTN or WIT seems much more opportunistic (this is especially true for the 'smaller' experiments). This is caused by the freedom of the tool in resource allocation. In a real application capricious resource allocation is undesirable: barge capacity has to be hired, and this kind of allocation can seriously damage the relation between the barge operator and the barge owner. Long term and sustainable resource allocation is required. A real world implementation of the tool must deal with this.

Figure 41 Container Transport Performance per Barge Operator per Experiment (BOP/SOP)

In BOP a transport booked at one barge operator (the source) can be executed by a different barge operator (the executor). Figure 42 shows this 'shift' in booked and delivered work for experiment E[96595]. It shows that the BOP causes a significant shift in work load per barge operator. CTN jumps out, this barge operator uses more foreign capacity for transportation than it uses its own. The results of the other experiments are similar.

| | Transport Source | | | |
Barge Operator	BCT	CTN	CTT	WIT
BCT	94,790	25,717	13,982	
CTN	20,394	42,980	19,083	9,093
CTT		5,886	112,398	23,925
WIT		26,357	24,291	87,940

Figure 42 Performance (TEU*KM) per transport source and executing barge operator E[96495]

Alternative Transportation

Alternative transportation is typically only used when the barge operator has no choice (because of the higher costs involved in the use of such capacity). Figure 43 shows the effect of bundling on the use of alternative transportation. Bundling reduces the need for alternative transportation and more is transported by barge; this confirms the 2nd hypothesis in section 4.5. Only in the busiest scenarios some alternative transportation is needed. This is due to the fact that the algorithm is much less constrained (no organizational constraints) in the allocation of resources.

Assuming that bundling allocates resources in an optimal way, barge capacity is not sufficient to meet the demand for transportation in the busiest cases (2065 and 2200 container transports) because some of the performance is still delivered using alternative transportation. Hence allocating additional barge capacity in these cases could reduce the overall transportation costs.
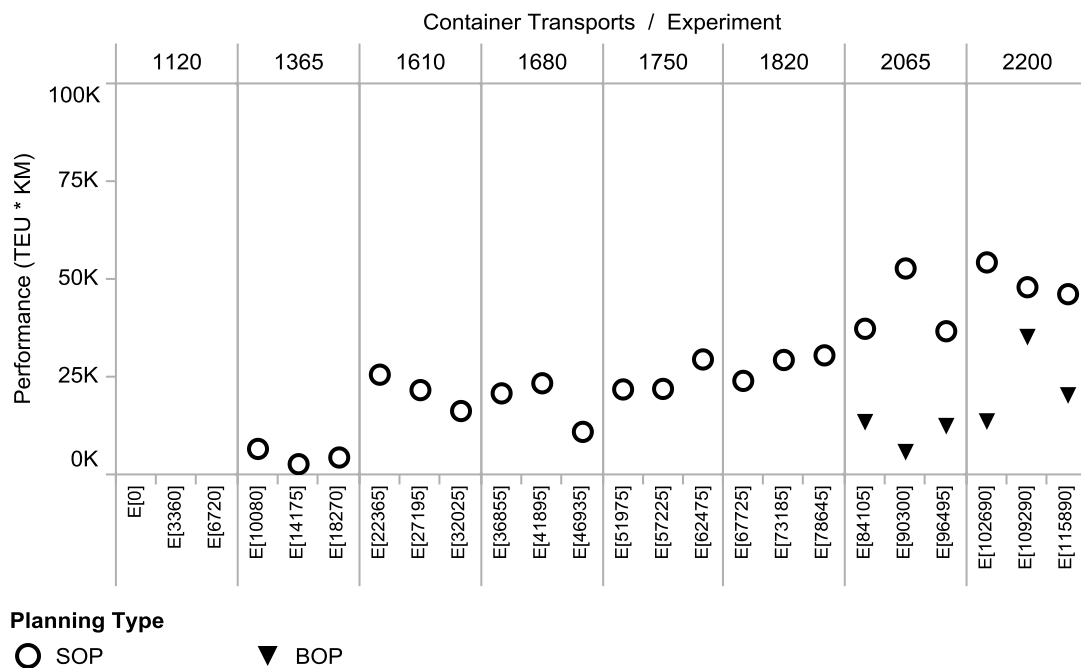
Figure 43 Barge Performance versus Alternative Performance per Experiment (BOP/SOP)

Barge Capacity

The previous sections showed that more is transported by barge and that work is assigned differently among operators; the allocation and use of barge capacity is different.

Figure 44 clearly shows that in the quieter cases (up to 400000 TEU*KM) the BOP approach allocates less barge capacity than the SOP approach; in busier cases this effect is smaller. Note that the BOP approach does more with the same capacity: this illustrated by the cases in which than 1900 TEU of barge capacity is allocated, SOP planning manages only to transport up to 500 K TEU*KM whereas BOP planning transports up to 550 K TEU*KM using the same capacity.

Using the BOP approach the use of resources starts at a lower point because barges can be used in a more flexible way; thus needing fewer barges in order fulfill the need for transportation. The use increases in a steeper way but generally stays below the distance covered for the same barges in the SOP variant while delivering the same performance.

The trends in both the barge capacity and the barge distance show that the BOP approach uses and allocates capacity in a more efficient way; BOP generally uses less capacity in order to deliver the same performance. This confirms the 1st hypothesis in section 4.5.

Planning Type
▼ BOP   ○ SOP
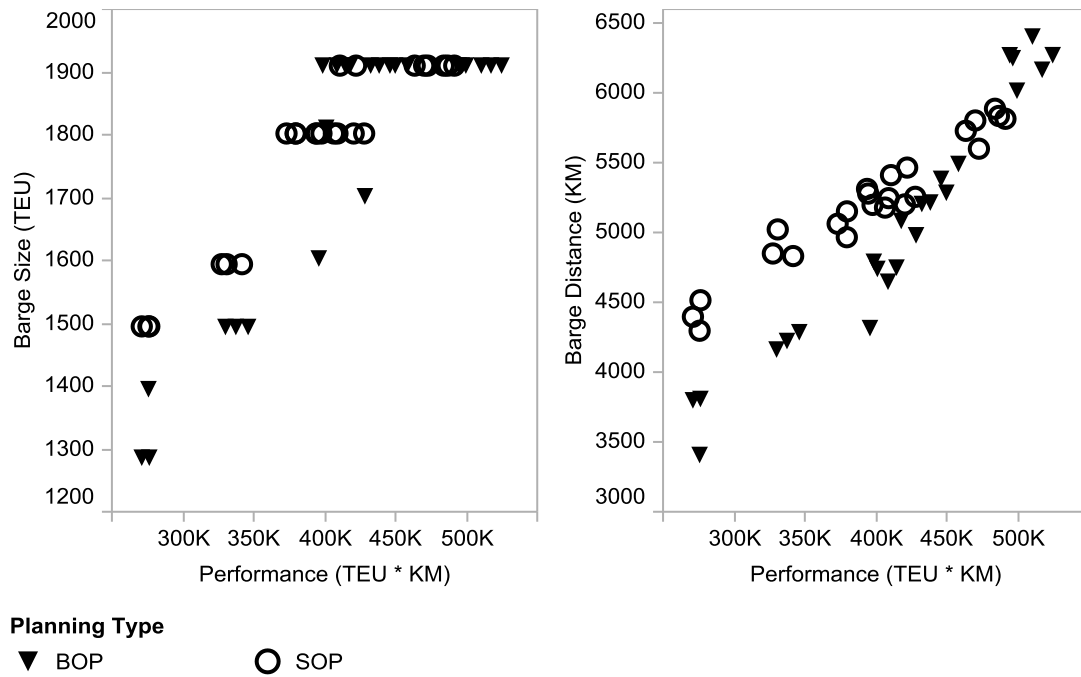
Figure 44 Barge Size and Barge Distance per Performance (SOP/BOP)

7.2.2   ANALYSIS OF BUNDLED OPERATOR PERFORMANCE

This subsection shifts from the analysis of the plans to the analysis of the effects the plan will have on the performance of the HCSS. This analysis focuses on the business critical aspects of the performance: costs and QoS. This subsection provides a big part of the answer to the last research question, because it gives detailed insight into the effects of Bundled Operator Planning.

Costs

Cost reduction is one of the main motivations for bundling cargo (requirement 1, see 6.2.1). The goal of a barge operator is to minimize costs; this will allow the operator to run a sustainable business and have a competitive advantage over other operators. Note that in section 4.4.2 the costs for transportation by barge have been discussed, this is not always case in this section because some of the comparisons will hard to interpret that way.

The overall costs per performance using the BOP approach differ heavily from the cost made using the SOP approach (see Figure 45). The costs per performance for SOP gradually decrease as the overall delivered performance increases. The costs per performance for BOP remain rather stable at ~0.27 Euro / TEU / KM. The trends for the BOP and SOP variants converge towards the end of the performance spectrum (for example the difference in costs between SOP and BOP for experiment E[109290] is less than 2%); the BOP and SOP approaches become almost interchangeable. However in these cases both the SOP and BOP variant have no more spare barge capacity and have to resort to the use of alternative transportation. BOP has proven to be more efficient than SOP in the allocation of barges, adding an additional barge to the planning problem in this case will probably lead to lower costs using BOP. Using the BOP approach significant costs reductions are possible, especially up to the point that all transports can be delivered by barge (in this case 450 K TEU * KM).

Planning Type
▼ BOP   ○ SOP
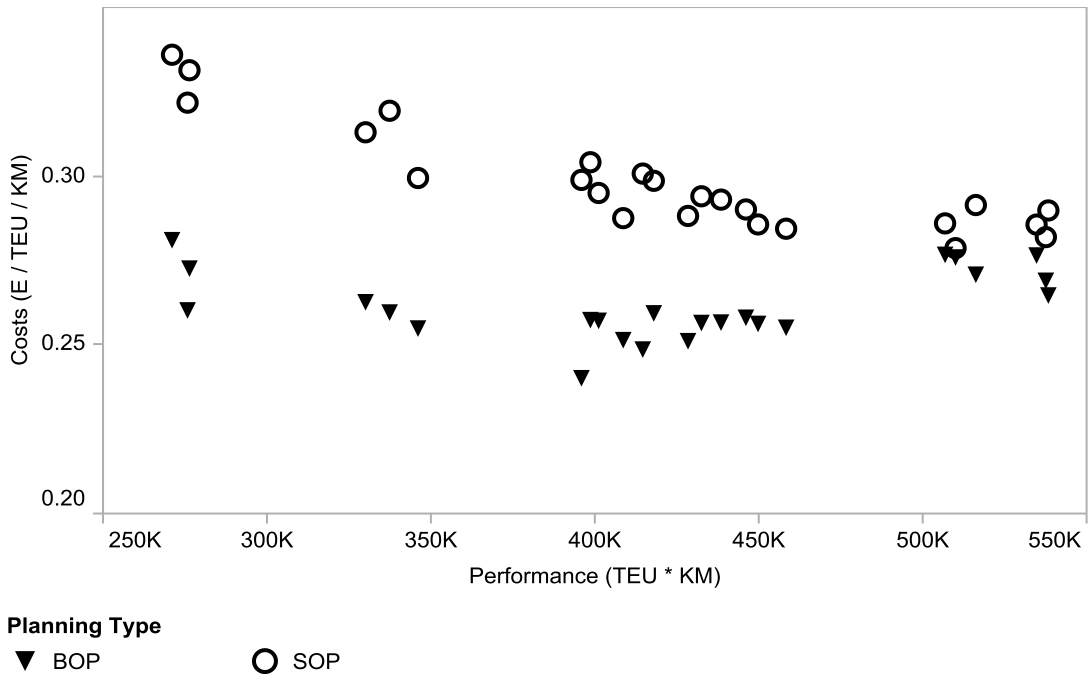
Figure 45 Costs per Performance (BOP/SOP)

When looking at the barge costs per barge operator comparing the SOP and the BOP approaches (see Figure 46) a number of things become clear:

1. The costs for BOP stabilize as the performance increase. This is caused by the different way resources are allocated, the position and other properties of the barges used play a much more important role in BOP resource allocation.
2. The costs using the BOP approach are much more scattered than the costs made using the SOP approach. The reason for this, again, lies with the different way of resource allocation. An extreme example is CTN, the costs for this barge operator lie between 0.23 Euro / TEU / KM and 0.42 Euro / TEU / KM (in this case transportation by truck is a viable alternative). This is caused by the substitutability of CTN's resources by those of other barge operators.


Barge Operator
○ BCT   □ CTN   + CTT   ✕ WIT
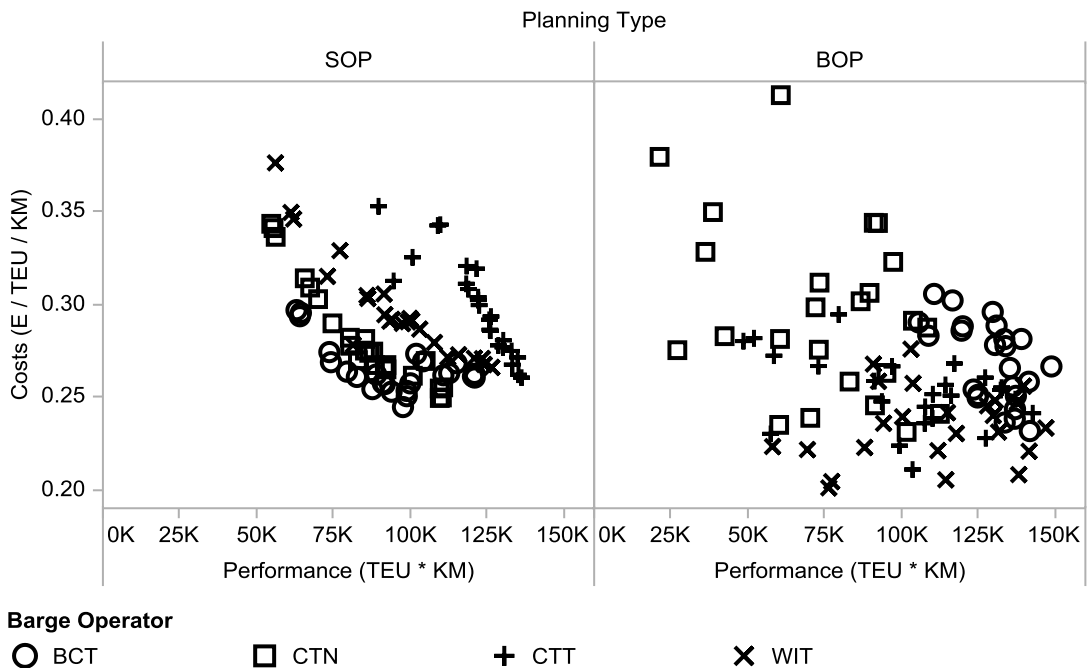
Figure 46 Costs per Performance per Barge Operator (BOP/SOP)

In 14 out of 24 plans bundled operator planning performs has a larger cost deviation from the planned cost than single operator planning (see Figure 47[14]). In nine cases the deviation is less, and in seven it improved because BOP avoided the double bookings made in SOP (see section 4.4.2 for more on this). In one case the deviation tied. This shows that the planned costs for bundled plans are generally a worse predictor for the executed costs than in unbundled plans; as the 19th century German field marshal Helmuth von Molkte would have said: *"No battle plan survives contact with the enemy."*
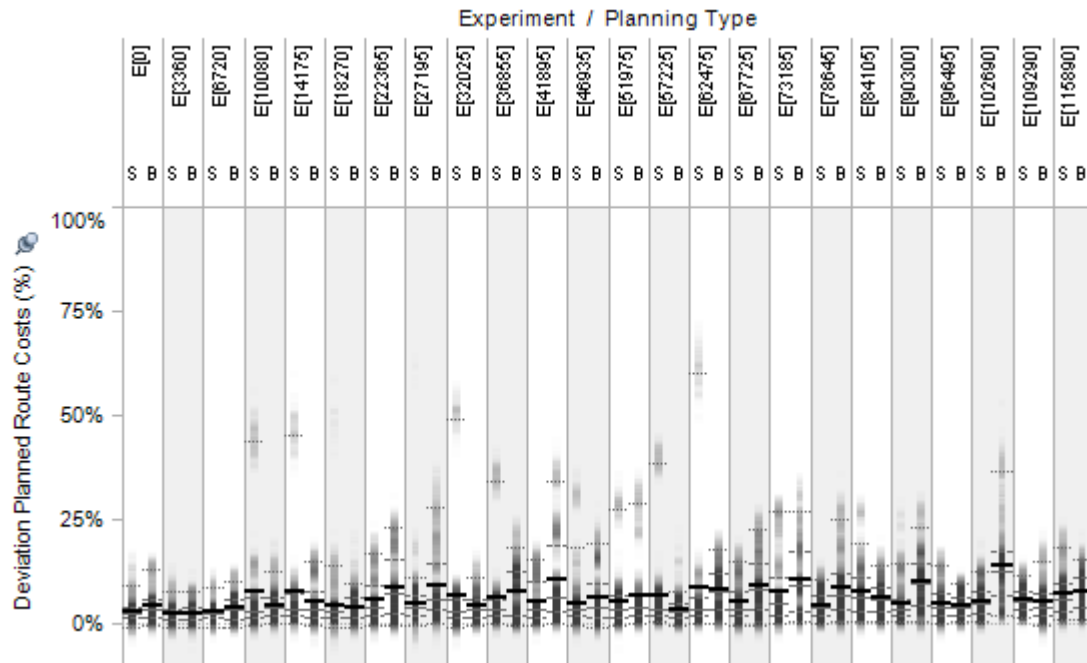
Figure 47 % Deviation from Planned Route Costs per Experiment (BOP/SOP)
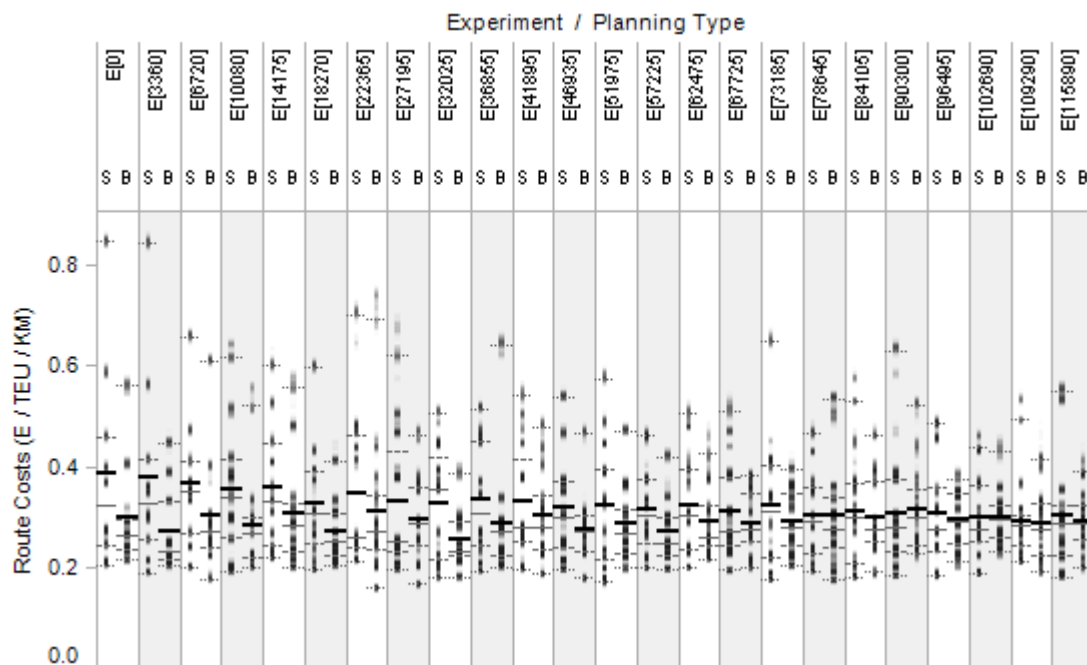
Figure 48 Route Costs (E / TEU/ KM)

---

[14] The horizontal markers depict the statistical properties of the distribution per experiment (thick black marker = average, thin grey marker = median, thin dashed grey marker = 25% or 75% percentile, thin grey dotted marker = 5% or 95% percentile).

Figure 48 show the costs per performance per route. This shows that different routes have very different costs (hence the normalization by means of performance). Deviations occur both in the SOP and the BOP approaches, however the average costs and the spread of the costs is generally smaller in BOP cases. This means that the financial risks involved in executing a BOP planning are generally smaller and that BOP planning is at least as robust as SOP planning from a costs perspective.

It also shows that effect of bundling in experiments in the last six experiments is marginal (> 1820 containers). This is due to lack of barge capacity. However, the experiments have been biased towards planning windows with higher complexity, there is a 5% chance than a window contains more than 1820 transport requests; bundling is effective in 95% of all cases.

Quality of Service

When service levels change due to bundling, the attractiveness of transportation by barge can change in the eyes of the customer, the shipper. The service level is measured by the difference in promised time of arrival of a contain transport and its actual time of arrival; the bigger this difference the worse the service level.

In 17 out of 24 experiments containers are on average delivered later using BOP planning (see Figure 49). In 5 experiments containers are on average delivered earlier. There are 2 ties. The use of BOP results in a worse service level, especially when looking at the 95th percentile (the top dotted gray lines in Figure 49). Bundled Operator Planning currently violates the robustness requirement set for bundling (see section 6.2.1) because of deteriorating the service levels.
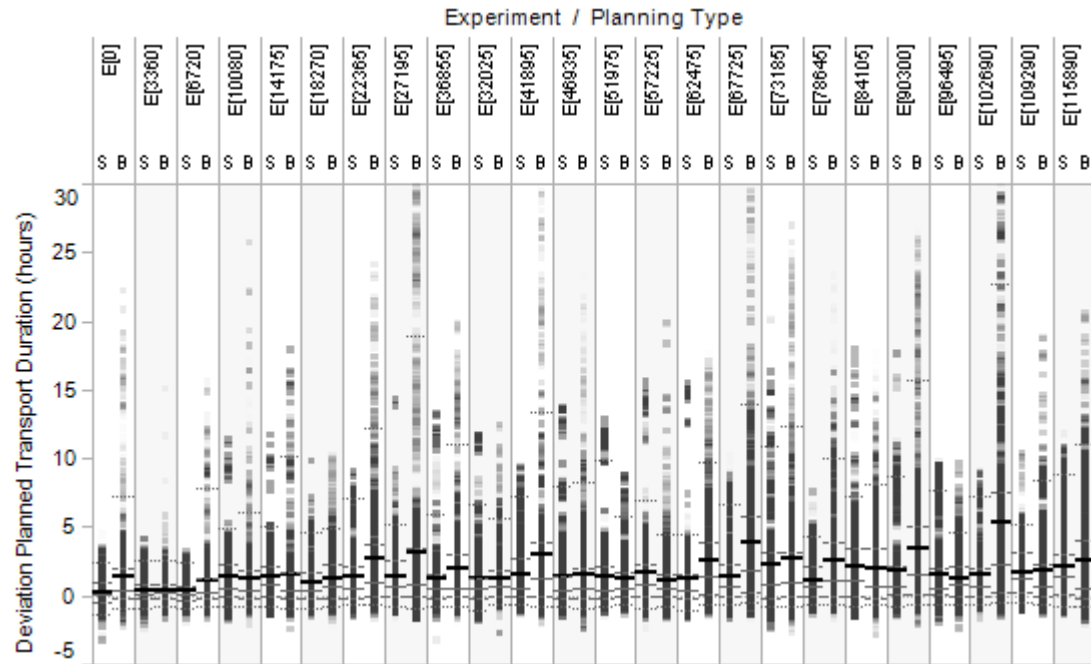
Figure 49 Deviation from Planned Transport Duration (hours) per Experiment (BOP/SOP)

The deterioration of the service levels is probably caused by the increased complexity of the plans produced; plans become riskier. The cause of these risks can usually not be influenced; however during the planning process (and in a planning algorithm) a number of measures can be taken in order to deal with the results of these risks:

1. Adding slack to the plans. In this case some slack is added after each activity in order to deal with uncertainties. The amount of slack should be dependent on the duration of the activity and on the risk taken during the activity. Adding slack is simple to implement, however it increases the transportation costs, as it takes longer to deliver the same performance.
2. Avoiding the coupling of high risks activities, i.e. calling at multiple unreliable terminals and taking unreliable routes. This should be incorporated in the planning process (and algorithm) by using information on the risks of the activities.

7.3 FEASIBILITY OF BUNDLING

In this section it is determined under which circumstances bundling is feasible in a multi-operator environment and which measures can be taken in order to make it feasible. Feasibility is discussed using the following subjects: requirements, cost attribution, resource allocation and trust.

Requirements

This might seem a bit superfluous but it is absolutely essential that the a proposed bundling design must meet the requirements of the operators who are bundling. The bundling implementation has to comply with the requirements set in section 6.2.1. Table 11 shows how and to what extent the current bundling PoC complies the requirements set.

Table 11 Requirements compliance Bundling PoC

	Name	*Compliance*
1	Costs	The costs of bundling opposed to the current situation have been evaluated in the previous section. Bundling can reduce costs significantly (up to 30%) in the right situation. However the cost advantage diminishes as experiments get 'larger' (i.e. more containers). The PoC complies with the cost requirement. However the requirement also notes that all operators involved should gain from bundling. This has not been addressed yet and will be discussed in the subsection on cost attribution.
2	Robustness	The robustness of the bundling PoC has been tested in the previous paragraph. Both the costs and QoS have been evaluated for robustness. Whereas the costs were in fact robust enough, the QoS was not. The QoS deteriorated for a significant number of the experiments using the bundling PoC. *The PoC does not comply with the robustness requirement.* In order to comply the suggested changes to the algorithm can implemented.
3	Impact	By using a line bundling network the impact has been limited because no large investments are required and the current way of working does not have to change. In the previous two sections it became clear that the impact of bundling might be even bigger than anticipated. Bundling changes the way planning is conducted: this results in rather differently structured plans and in capricious resource allocation. The former will take some getting used to whereas the latter can have serious consequences on the long term. The impact of bundling will only be small enough when the resource allocation is properly conducted (this is discussed in subsection on resource allocation).

As such the bundling PoC basically complies with two out of three requirements. The PoC had to prove that bundling was possible, and as such it did by creating feasible and cost effective bundled plans. However the plans (and therefore bundling) should only be considered feasible when the robustness is improved.

Picking up the Tap

Every operator involved in bundling should gain from it (Requirement 1 - section 6.2.1). Reality draws a different picture: some operators gain by bundling whereas others actually lose, these operators are picking up the tap for bundling. There is no incentive for a barge operator to get involved in bundling when he does not gain from it; loosing operators must compensated in order to keep them in a bundling alliance.

The reason that the one operator gains more from bundling than the other operator is determined by the location of the operator's terminal(s) in the network and by the container flows within the network. The more interconnected a terminal is, the more opportunities for shared barge usage. The more the flows of an operator can share parts of their routes with other flows, the more opportunities exist for resource sharing. These are intrinsic advantages of certain barge operators.

There are both operational and structural costs a 'loosing' barge operator needs to be compensated for. The operational costs that an operator needs to be compensated for are the costs made that operator when transporting containers belonging to another operator. These costs are different per plan. Compensation can be quite straightforward when only the costs for transportation are taken into account. Figure 50 shows the cost attribution table for experiment E[41895], in this experiment for example CTN uses BCT's resources for transportation and has to compensate BCT for € 7949.

Planning Type	Barge Operator	Transport Source				Grand Total
		BCT	CTN	CTT	WIT	
SOP	BCT	€23,327				€23,327
	CTN		€23,741			€23,741
	CTT			€48,452		€48,452
	WIT				€29,266	€29,266
	Total	€23,327	€23,741	€48,452	€29,266	€124,787
BOP	BCT	€18,007	€7,949	€3,826	€3,579	€33,362
	CTN	€3,219	€5,463	€5,554	€5,919	€20,154
	CTT		€2,195	€20,558	€700	€23,452
	WIT		€9,552	€11,210	€10,642	€31,403
	Total	€21,226	€25,158	€41,147	€20,840	€108,371

Figure 50 Cost (E) attribution table for Experiment E[41895] (SOP/BOP)

More complex costs calculation are possible, which for example could take the detours required for transporting the container into account. In such cases the use of a profit sharing scheme is a promising approach.

The structural problem is more subtle and insidious. Although the overall costs for transportation decrease as a result of bundling, it may cause an increase for a barge operator's individual costs. This illustrated by CTN's costs in Figure 50 (this is a reoccurring cost pattern for CTN). In this case only compensation for the transport costs will not do, and more is needed. In the case as presented Figure 50 it is quite easy to uncover the minimum compensation CTN requires: € 1417. When implementing bundling the unbundled plan will usually not be available; making the previous calculation impossible.

Feasible bundling requires a 'fair' allocation of both structural and operational costs; a profit sharing scheme can be used to accomplish this. Unfortunately there is little literature on this. Most of the suggested schemes use cooperative gaming theory as the basis for profit sharing (Engevall, Göthe-Lundgren et al. 2004; Krajewska, Kopfer et al. 2008). Such a scheme should be heavily embedded in an institutional design because matters of costs and profit lie at the core of the operators involved and disputes on such matters will easily arise. The application of an effective profit scheme is therefore an important avenue for further research.

Resource Allocation

In the previous section it became clear that the bundling changes the way resource are allocated. The main problem encountered was that the capacity allocation for especially the smaller experiment sizes was rather opportunistic. Barge capacity is hired; capricious barge allocation can be extremely damaging for a barge owner on short term, this will disturb the (long term) relation between the owner and the operator. Feasible bundling requires a proper resource allocation mechanism.

It should be noted that the resource allocation method used during the experiments was extremely unconstrained. This was due to the fact that the algorithm constructed plans from scratch and because the different options for capacity expansion were relatively similar (making a choice between different barges random). In a real application the solution uses partial plans and has a more limited choice in selecting barge capacity. This will stabilize the allocation of resources. Further steps could be taken by adding more rules to the selection of capacity.

Another related problem with (mildly) unconstrained resource allocation is the fact that a barge operator loses control over its capacity. This can be undesirable. A barge operator should be able to reserve capacity for its own use, which bundling is not allowed to allocate for use by a different barge

operator. This limits the capriciousness of the resource allocation even more and it also limits transparency. Such an arrangement must be a part of institutional design because guarantees need to be made about the available capacity.

Trust

Trust between barge operators is essential for feasible bundling. Barge operators have to coordinate their actions and will deliver containers for each other in order to improve the efficiency of hinterland container shipping. The more trust there is between barge operators, the easier bundling becomes. Trust can be created in a number of ways:

1. An institutional design can be used to protect and enforce the arrangements for bundling. This should describe the rules for engaging in bundling, rules for resource allocation, rules for revenue management, rules for entering and exiting a bundling alliance and finally rules for abuse. This determines how operators should behave, and it will punish strategic and opportunistic behavior by barge operators.
2. The platform which enables bundling, a.k.a. the bundling implementation, can take a number of measures that create trust in bundling: by limiting the exchange of information (reducing transparency) and by creating robust and efficient plans.
3. A roadmap that takes care of the implementation of bundling. The roadmap takes a number of operators through a number of steps in which bundling is implemented. The roadmap has three consecutive goals: create awareness of bundling, create trust in bundling and finally create commitment to bundling.

7.4 CONCLUSION

Bundled operator planning as a new planning paradigm has been evaluated over the course of this chapter. This has been done by comparing the differences between bundled operator planning and single operator planning (see chapter 4) and by evaluating the side effects of bundling. The synthesis of this evaluation answers the last research question:

> 4. *What are the costs of hinterland container shipping using bundled operator planning?*

The following costs and (side) effects of bundled planning have been have been observed during the evaluation:

1. The most notable effect on the planning process is the acceptability of the plans created by the PoC are very different from the plans created by planners. Planners are currently less inclined to accept such plans. More acceptable plans can be created by adding constraints to the planning algorithm.
2. The costs per performance for bundled plans remain rather stable as performance increases (~0.27 E / TEU / KM). This differs heavily from unbundling plans where costs gradually decrease as performance increases. The trends for the BOP and SOP variants converge towards the end of the performance spectrum; the BOP and SOP approaches become almost interchangeable. The deviations in costs are similar for bundled and unbundled plans. Bundled operator planning is cheaper than single operator planning in 95% of all cases.
3. Bundled plans usually have a worse service level than unbundled plans; container reach their destination much later than promised more often than in the single operator variant. Bundled operator planning currently violates the robustness requirement for bundling and this needs to be improved.
4. The allocation of resources in bundled operator planning will be much more opportunistic than in single operator planning. Opportunistic barge allocation can damage the relation between the barge owner and barge operator. The allocation of barge capacity is controlled by the barge operator and a planning algorithm should respect this.
5. The cost attribution shows that not every barge operator gains from bundling on an operational level (lower transportation costs). All barge operators should gain from bundling, therefore some sort of a profit sharing scheme is needed.

Bundling operator planning can be implemented within the HCSS planning process and creates plans that are different yet more cost effective than single operator plans. Bundled planning is an improvement in all but the most complex cases. However, the robustness of these plans must improve, rules for resource allocation must be established and all barge operators should gain from bundled planning.

8 CONCLUSIONS & RECOMMENDATIONS

This chapter describes the conclusions and recommendation which can be made based on the outcomes of the research. The conclusions are the answers to the research questions. The questions will be revisited in section one, the synthesis of these questions answers the main research question. The recommendations describe future avenues for research and the potential implications of the outcomes of the research. The recommendations will be described in section two.

8.1 RESEARCH QUESTIONS REVISITED

The research questions have been answered during the research, they are revisited during this section. All the research questions targeted an aspect of the main research questions, by synthesizing these aspects the main research question is answered.

1. What are the costs of hinterland container shipping using the current, single operator, planning method?

The purpose of this first research question was to understand the costs of the current planning approach and to set a goal for a (future) bundled operator planning approach. The results of the future planning method should be an improvement of the current costs for the planning method to be effective. The main conclusions are listed below:

1. The use of alternative transportation increases steadily as the levels increase. The source for alternative transportation (mainly by truck) is CTT. CTT does not currently allocate more barge capacity than the 515 TEU provided by the four barges it uses. The additional demand for performance is handled by truck. The ceiling in CTT's barge performance is mainly caused by the limitation imposed by the Twente Canal, its distance to the terminals in the ports of Rotterdam and Antwerp, and the lack of additional barges. The other three barge operators (BCT, CTN, and WIT) hardly require any alternative transportation. These operators have the luxury of having a better match between demand for transportation and the supply of barge capacity.
2. The costs per kilometer per standard unit are decreasing steadily as the levels and transport performance used in the experiments increases. This is caused by the fact that barges can be planned in a much more efficient way when large volumes of containers belonging to similar flows are available for transportation..
3. The execution of the plans yielded that in most cases the costs are underestimated and that the overall deviation in costs is high. The deviations in costs are caused by the use of too little slack in creating the plans or the planning or by the coupling of high risk activities.
4. The duration of transports (which is a measure for QoS) increases as the size and the performance of the experiments increase. The deviations in the duration of transports are high; it can only be guaranteed that 95% of the transports are delivered less than 12 hours late.

2. Which bundling stereotypes are suitable for container hinterland shipping?

The actors selected for bundling are the four separate barge operators internal to BCTN: BCT, CTN, CTT and WIT. All regular container transports between the inland terminals and the seaside terminals are to be bundled on the joint capacity of the four selected operators.

There are four bundling network stereotypes: Line, Hub-and-Spoke, Trunk-Line Feeder and Trunk-Line collect and distribute. Line bundling is the only stereotype that does not require a transshipment terminal. Transshipment terminals are problematic because of a number of reasons: CTN is extremely limited in terms of space and is the only terminal suited for the role of transshipment terminal, the investments and commitment for creating a new terminal are high and transshipment causes extra transportation costs. The leaves line bundling as the only feasible bundling network stereotype. The low impact and potentially easy win in costs appear to make line bundling a good fit for the HCSS.

3. What bundling algorithms can be used in the implementation of a bundling tool?

The planning problem the algorithm needs to solve is a version of the pickup and delivery problem with split loads (PDPSL). Groups of containers with similar properties are seen as separate loads. These loads need to be picked up at one terminal and transported to another terminal without violating constraints. These loads can be transported using one or more barges.

Because of the size and the complexity of the bundled planning problems an automated planning approach (using an algorithm) is preferred. The pickup and delivery problem with split loads is NP-hard. This means that exact algorithms can be ruled out as feasible algorithms. There are numerous highly competitive algorithms which can provide near-optimal solutions to the pickup and delivery problem with split loads. The Adaptive Large Neighborhood Search has been selected because of its highly

flexible nature. This algorithm can be easily adapted to the unique challenges hinterland container shipping offers.

4. What are the costs of hinterland container shipping using bundled operator planning?

The following costs and (side) effects of bundled planning have been have been observed during the evaluation of bundling:

1. The most notable effect on the planning process is the acceptability of the plans created by the PoC are very different from the plans created by planners. Planners are currently less inclined to accept such plans. More acceptable plans can be created by adding constraints to the planning algorithm.
2. The costs per performance for bundled plans remain rather stable as performance increases. This differs heavily from unbundling plans where costs gradually decrease as performance increases. The trends for the bundled operator and single operator planning variants converge towards the end of the performance spectrum; the approaches become almost interchangeable. The deviations in costs are similar for bundled and unbundled plans. Bundled operator planning is cheaper than single operator planning in 95% of all cases.
3. Bundled plans usually have a worse service level than unbundled plans; container reach their destination much later than promised more often than in the single operator variant.
4. The allocation of resource in bundling operator planning can be much more opportunistic. Opportunistic barge allocation can damage the relation between the barge owner and barge operator. The allocation of barge capacity is controlled by the barge operator and a planning algorithm should respect this.
5. The cost attribution shows that not every barge operator gains from bundling on an operational level (lower transportation costs). All barge operators should gain from bundling, therefore some sort of a profit sharing scheme is needed.

Bundling operator planning can be implemented within the HCSS planning process and creates plans that are different yet more cost effective than single operator plans. However bundling comes at a price: the robustness of these plans is lower, rules for resource allocation can be more capricious and some barge operators do not gain from bundling. These issues must be addressed.

What bundling stereotypes and algorithms can be used to improve the efficiency of hinterland container shipping by means of bundling?

The main question has been answered by the research questions. Line bundling and the implementation of the Adaptive Large Neighborhood Search planning algorithm for the barge operators within BCTN have shown to improve costs in 95% of the cases. However, the robustness decreases. *The results of this research have made a strong case for bundling and have proven that bundling improves efficiency of hinterland shipping*.

8.2   RECOMMENDATIONS

Based on this research a number of recommendation can be made. These recommendations entail the introduction of potential measures in the HCSS, further research or roadmaps for further work. Most recommendations are typically actor bound, so they are described per problem owner:

BCTN

1. The bundling proof of concept constructed during this research could easily be adapted for the use within BCTN. The robustness needs of the results needs to be improved, but the other drawbacks do not apply to a single actor situation; making the implementation within BCTN very feasible and profitable.
2. BCTN should make an effort to integrate at least three out of the four Modality instances. This will make the current planning considerably easier. This is one of the reasons that current bundling opportunities are neglected. The integration of the four systems would make the implementation of a bundling tool easier.
3. During the research it was shown that the use of too optimistic planning constraints will lead to infeasible plans. This also was the case in both the current as the future situation. The planning constraints by BCTN should be (re)evaluated on a regular basis, in order to improve the quality of plans.
4. Further research should be conducted into the fuel efficiency of barges – there is little information available on this. By using more eco friendly barges, BCTN could further improve its position compared to that of operators using different modalities. It would also make BTCN eligible for government grants.

5. At the end of all research activities the news came in that BCTN will be operating a new terminal in Alblasserdam (BCTN 2009). This project is backed the RPA and is an attempt to relieve the stressed road infrastructure in the Rotterdam area. This has implications for the bundling network chosen, line bundling might not be optimal in this case. Further research should be conducted into this.

VITO

1. The major drawback of the current bundling PoC is its limited applicability in a multi-actor context. This is due to the fact that a profit sharing scheme and a resource allocation scheme need to be devised. Both could be embedded in an institutional design. The resolution of these issues makes bundling feasible within VITO.
2. On a related note a roadmap for the implementation should be created. By involving all required actors and creating a stable environment awareness, trust and commitment; which speeds up the implementation of bundling.

INITI8

1. The current implementation of the bundling PoC is not a complete bundling tool. The robustness, coordination capabilities and the transparency of the tool need to be improved. However, it is a start. The current research only focused on bundling on bundling between a limited number of ports and one type of transportation. All the tools used in the research can be easily adapted to different markets, to different transport types and to different actors. Other problems can be solved using the tool set. For example bundling could be implemented for a number of barge operators which are active on the river Rhine (in this case different bundling networks are feasible).
2. The last recommendation is that INITI8 can speed up the implementation of bundling by creating awareness for such problems by themselves. Cooperating with VITO and BCTN seems to be natural choice for this.

9   REFLECTION

This final chapter reflects on the research as a whole and tries to assess if the research is valid and if the research process has been fruitful. The goal of this chapter is to learn from the research project and to improve research skills. This chapter is the only chapter that has been written from the point of view of the author - a reflection always has a bias.

9.1   RESEARCH VALIDITY

This section discusses the validity of the research. Attention is given to the demarcation of the problem, the methods used to answer the research questions and the relevancy of the research.

Demarcation

The demarcation defines the system under consideration; the system is a model of reality. A model can be seen as a theory of reality. Multiple valid models of the same system can coexist, there is no perfect model (Saleh 2000). Differences between models are caused by the question the model tries to answer, the modeling method used, the means available and the preferences of the modeler(s). This means that it should be investigated whether the system chosen was a correct and valid representation of reality. This has been partially answered by verification, validation and evaluation steps during the research (sections 3.7, 4.4, 6.3.5 and 7.2). However these steps only checked if the results and behavior were valid within the demarcation given.

The demarcation chosen provided a representation of the BCTN's regular services between their inland terminals and seaside terminals. It was chosen in such a way that the effect of applying bundling to BCTN as one barge operator (instead of four) could be evaluated.

The chosen demarcation was perhaps too broad. The analysis was very involving due to complexity caused by the number of objects in system and the amount data available on the system. For example it took over a month to gather and to merge the data of multiple terminals. This could have saved a lot of time (up to 10 weeks!). Another unintended effects was that some interesting system elements were underexposed, for example: the effect of changes in rotation plans (section 1.1) and the use of waterways by barges. A more limited demarcation, for example only two barge operators, could have prevented this.

CTT was taken into account. However only aggregated data on the processes of CTT was available. The detailed data of the other barge operators was available though. The size and scope of CTT operations has been validated, however this is not the same as the use of the true data.

Methodology

The methodology is the path taken in order to answer the research questions. These could perhaps have been answered different way. A bundling PoC was the goal of the research, however the research questions could have been answered using a more top-down approach in which some tasks could have been omitted (such as the materialization of the PoC) because the effect could have been approximated using a number of static calculations. However the approach chosen shows that bundling is applicable and is not mere conjecture of applicability.

Too much attention was paid to the current situation, less comprehensive analysis would have been sufficient to answer the main research question. However, this only was proven by the path taken. Such insight can only be inferred in hindsight and proves the adage: "*When you've got a hammer, everything looks like a nail.*"

The current research has a high degree of internal consistency. The PoC has been tested using simulation, Moonen (2009) shows that simulation has a high degree of rigor but low inherent relevancy. Relevancy can be improved by the different and independent methods. For instance testing the PoC in a setting in which BCTN or others actually use the it. The current research only shows the maximum effectiveness, whereas tests (and use) in practice will determine the true effectiveness of bundling. This must be tested, and is an important avenue for further research.

Relevancy

The research showed that bundling (given a few requirements) is applicable to the HCSS. The bundling proof of concept can be extended and serve as basis for a true implementation. This is both relevant for INITI8, BCTN and even VITO as problem owners; the research is relevant from a practical point of view.

The research is one of only three sources (to the best of the authors knowledge) describing how to effectively solve a Pickup and Delivery Problem with Split Loads. The adaptation of the ALNS heuristics to work on multiple levels (calls and transports) is a new concept. There is little literature on the scalability of planning algorithms, this research showed that this can be achieved easily. However, the research also showed some voids in current research:

1. Only a small body of knowledge existed on both horizontal cooperation between actors. It is known that such coordination can be beneficial to the parties involved. However, little is known on decent profit sharing schemes which is critical for successful cooperation.
2. Another avenue of research is dealing with uncertainty in plans while still planning. There is some literature on this, for example the vehicle routing problem with stochastic demands. The current solutions cannot deal with intricate problems such as the planning of hinterland container shipping. It is interesting to see which algorithms are usable and what their effect would be.

The research provided scientifically relevant results and insights.

9.2 RESEARCH PROCESS

The section reflects on the research process. It tries to answer if the research process has been effective. The effectiveness of the research project is a balance between the result the research yielded and the turnaround of the research.

The following deliverables have been created over the course of the research: a simulation model of the HCSS, a barge operator datawarehouse, a map with all major hinterland shipping connections and terminals, an analysis of the current situation, an algorithm for bundling cargo, the design and implementation of a bundling PoC, an evaluation of a future situation in which bundling is applied and this report.

The turnaround of the project can characterized very well using the following quote by Douglas Adams (1952-2001): *"I love deadlines. I like the whooshing sound they make as they fly by."* Almost all deliverables were late, especially the thesis chapters. In all the execution of the research too twice as long as originally planned. This was caused by a number of reasons:

1. Unrealistic and bad planning on behalf the author. The duration of almost all activities has been seriously underestimated. A plan has to be adjusted while its being executed, but the author decided to stick to the original goals and not to change the level of detail accordingly. This is especially bizarre since one of the topics of the research was planning.
2. To little progress evaluation. The author was very reluctant to deliver unfinished deliverables. A deliverable is never truly finished; intermediate evaluation could have saved time.
3. The research goals were set high and should have been adjusted in order to keep the turnaround in check. This is (again) the responsibility of author.

9.3 CONCLUSION

The result of the research are both practically and scientifically relevant, are valid and proved that the application of bundling to the HCSS is feasible. However, the demarcation used was probably too broad, different methods could have been use and actors should have been involved more.

The quality and the amount of deliverables created during the research was good, however this came at a price; the research took more than twice as long as planned, which is in line with the first supervisors adage that an interesting project always takes twice as long as you think, even when you account for it.

LITERATURE

Adams, D. (1952-2001). "I love deadlines. I like the whooshing sound they make as they fly by." from http://en.wikiquote.org/wiki/Douglas_adams.

Andel, M. C. v. (2007). Measures for Improving VITO Container Barges Handling at ECT Delta, Port of Rotterdam. Delft, TU-Delft.

BCTN (2008). Container Transport. BCTN.Process.plaatje.bmp. Nijmegen, BCTN. **1975 KB:** Container Transport Process.

BCTN. (2009). "BCTN gaat containertransferium Alblasserdam exploiteren." Retrieved 2009-09-08, 2009, from http://www.bctn.nl/nl/newslog/2009-06-29/bctn-gaat-containertransferium-alblasserda.

BCTN. (2009). "Binnenlandse Container Terminal Nederland - Home." Retrieved 2009-05-19, 2009.

Berbeglia, G., J.-F. Cordeau, et al. (2007). "Static pickup and delivery problems: a classification scheme and survey." TOP: An Official Journal of the Spanish Society of Statistics and Operations Research **15 (2007)**(1 (July)): 1-31.

Binsbergen, A. v. and J. Visser (2001). Innovation Steps Towards Efficient Goods Distribution Systems for Urban Areas, Efficiency Improvement of Goods Distribution in Urban Areas. Faculty of Civil Engineering and Geosciences. Delft, Delft University of Technology. **PHD:** 576.

Bots, P. W. G. (2002). Introduction to Systems Engineering and Policy Analysis, A practical guide to systematic problem solving. Delft, TU-Delft: Faculty of Technology, Policy and Management.

Bräysy, O. and M. Gendreau (2005). "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms." TRANSPORTATION SCIENCE **39**(1): 104-118.

Brooks, F. (1975, 1995). The Mythical Man-Month: Essays on Software Engineering Addison-Wesley.

Campbell, A. M. and M. Savelsbergh (2004). "Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems." TRANSPORTATION SCIENCE **38**(3): 369-378.

Caramia, M. and F. Guerriero (2009). "Aheuristic approach to long-haul freight transportation with multiple objective functions." Omega **37**(3): 600-614.

CBRB (2003). Basisdocument Containerbinnenvaart. Rotterdam, CBRB**:** 56.

Centraal Bureau voor de Rijn- en Binnenvaart (2003). Basisdocument Containerbinnenvaart. Rotterdam, Centraal Bureau voor de Rijn- en Binnenvaart**:** 56.

Connekt (2003). Truck Afhandeling, Eindrapport FAMAS.MV2 - Project 12.1. Delft, Connekt**:** 134.

Cordeau, J.-F., G. Laporte, et al. (2008). Recent Models and Algorithms for One-to-One Pickup and Delivery Problems. Vehicle Routing: Latest Advances and Challenges. B. L. Golden, S. Raghavan and E. A. Wasil. Boston, Kluwer**:** 35.

Crainic, T. G. (1999). "Service Network Design in freight transportation." European Journal of Operations Research **122**(2000): 16.

CTT (2008). Vorschlägen zur Leercontaineroptimierung eines Inlandterminals. Hengelo. **BSc:** 62.

Davidsson, P., J. A. Persson, et al. (2007). On the Integration of Agent-Based and Mathematical Optimization Techniques Agent and Multi-Agent Systems: Technologies and Applications. Springer. Berlin / Heidelberg, Springer **Volume 4496/2007:** 1-10.

Defense Acquisition University Press (2001). Systems Engineering Fundamentals. D. A. U. Press. Fort Belvoir, Defense Acquisition University Press**:** 51-53.

Dietz, J. L. G. (2006). Enterprise Ontology - Theory and Methodology. Berlin Heidelberg, Spring Verlag.

Douma, A., P. Schuur, et al. (2006). Applying revenue management to agent-based transportation planning. Beta Working Papers. Twente, University of Twente**:** 24.

Douma, A., P. Schuur, et al. (2007). Barge rotation planning and quay scheduling in the port of Rotterdam. Enschedue, University of Twente, School of Management and Governance**:** 32.

Douma, A. M. (2009). Aligning the Operations of Barges and Terminals through Distributed Planning. Beta Research School for Operations Management and Logistics. Enschede, University of Twente. **PHD:** 255.

Eggermond, M. A. B. v. (2006). Specification of the Chemical Tanker Simulation Model. TU-Delft: Faculty of Technology, Policy and Management. Delft, Delft University of Technology. **BSc:** 72.

Eggermond, M. A. B. v. (2007). Consumer choice behaviour and strategies of air transportation service providers. Faculty of Civil Engineering and Geosciences – Transport & Planning Technology, Policy and Management. Delft, Delft University of Technology. **MSc.:** 228.

Engevall, S., M. Göthe-Lundgren, et al. (2004). "The Heterogeneous Vehicle-Routing Game." TRANSPORTATION SCIENCE **38**(1): 71–85.

Freeman, E., E. Robson, et al. (2004). Head First Design Patterns Sebastopol, CA, O'Reilly.

Goldberg, D. E. (1989). <u>Genetic Algorithms in Search Optimization and Machine Learning</u>, Addison-Wesley.

Goor, A. v. and W. Ploos van Amstel (2002). <u>Fysieke distributie, Denken in toegevoegde waarde</u>, Stenfert Kroese.

Hapag-Lloyd (2008). EU3 Loop. 20080915.Hapag-Lloyd.EU3-Europe-Asia-Loop-3.gif. Hamburg, Hapag-Lloyd. **19 Kb:** Overview of Hapag-Lloyds EU3 Loop.

Hassin, R. and S. Rubinstein (2005). "On the complexity of the k-customer vehicle routing problem." <u>Operations Research Letters</u> **33**: 71-76.

Hemmelmayr, V. C., K. F. Doerner, et al. (2009). "A variable neighborhood search heuristic for periodic routing problems." <u>European Journal of Operational Research</u> **195**: 791–802.

Hillier, F. S. and G. J. Lieberman (2005). <u>Introduction to Operations Research</u>, McGraw Hill.

Hjaltason, G. R. and H. Samet (2002). "Speeding up construction of PMR quadtree-based spatial indexes." <u>VLDB Journal</u> **2002**(11): 190–137.

Horst, M. v. d. and P. d. Langen (2008). "Coordination in hinterland transport chains: A major challenge for the seaport community." <u>Maritime Economics and Logistics</u> **10**: 108—129.

Inspectie Verkeer en Waterstaat (2008). Nulmeting naar dwarsscheepse stabiliteit in het containervervoer in de binnenvaart, Inspectie Verkeer en Waterstaat**:** 40.

Jacobs, P. H. M. (2005). The DSOL Simulation Suite. <u>TU-Delft: Faculty of Technology, Policy and Management</u>. Delft, Delft University of Technology. **PHD:** 228.

JBoss. (2009). "Hibernate."   Retrieved 2009-08-29, 2009, from https://<u>www.hibernate.org/</u>.

Kimball, R. (2004). Kimball Design Tip #51: Latest Thinking On Time Dimension Tables. <u>Kimball Design Tips</u>. Kimball Group.

Konings, R. (2006). Hub-and-Spoke Networks in Container-on-Barge Transport. <u>Transportation Research Record: Journal of the Transportation Research Board</u>. T. R. Board. Washington, TRB. **1936:** 23-32.

Krajewska, M. A., H. Kopfer, et al. (2008). "Horizontal cooperation among freight carriers: request allocation and profit sharing." <u>Journal of the Operational Research Society</u> **59** (11): 18.

Kreutzberger, E. D. (2008). "Distance and time in intermodal goods transport networks in Europe: A generic approach." <u>Transportation Research</u> **Part A 42 (2008)**: 20.

Krogt, R. v. d. (2007). Privacy Loss in Classical Multiagent Planning. <u>Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology</u>, IEEE Computer Society.

Laporte, G. (2009). Fifty Years of Vehicle Routing. Montreal, Canada Research Chair in Distribution Management HEC Montreal.

Law, A. M. and W. D. Kelton (1991). <u>Simulation Modeling and Analysis</u>. Singapore, McGraw-Hill, Inc.

Lu, Q. and M. M. Dessouky (2006). "A New Insertion-based Construction Heuristic for Solving the Pickup and Delivery Problem with Time Windows." <u>European journal of operational research</u> **1752**(2): 672-687.

Madahain, J., D. Fisher, et al. (2005). Analysis and Visualization of Network Data using JUNG. <u>Journal of Statistical Software</u>. **VV**.

Marcken, C. d. (2003). Computational Complexity of Air Travel Planning. <u>Public notes on computational complexity</u>, ITA Software**:** 51.

Mieghem, P. v. (2006). <u>Data Communications Networking</u>. Amsterdam, Techne Press.

Miller, R. (2003). "Practical UML: A Hands-On Introduction for Developers."   Retrieved 2003-12-03, 2009, from <u>http://edn.embarcadero.com/article/31863</u>.

Mitrovic-Minic, S. (1998). "Pickup and Delivery Problem with Time Windows: A Survey." <u>SFU CMPT TR</u> **12**: 43.

Modality Software Solutions. (2009). "Modality."   Retrieved 2009-09-15, 2009, from <u>http://www.modality.nl/</u>.

Moonen, H., B. van de Rakt, et al. (2005). Agent technology supports inter-organizational planning in the port. <u>ERIM REPORT SERIES RESEARCH IN MANAGEMENT</u>. Rotterdam, Erasmus Research Institute of Management (ERIM)**:** 21.

Moonen, H. M. (2009). Multi-Agent Systems for Transportation Planning and Coordination. <u>Erasmus Research Institute of Management</u>. Rotterdam, Erasmus Universiteit Rotterdam. **PhD:** 256.

Mundy, J., W. Thornthwaite, et al. (2006). <u>The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset</u>, Wiley.

Nagata, Y. and O. Bräysy (2008). Efficient Local Search Limitation Strategies for Vehicle Routing Problems. <u>Evolutionary Computation in Combinatorial Optimization</u>. Heidelberg, Springer Verlag. **4972:** 48-60.

Nance, R. E. (1995). <u>Simulation Programming Lanuages: An Abridged History</u>. Winter Simulation Conference.

LITERATURE

Navathe, S. B. and R. Elmasri (2006). Fundamentals Of Database Systems Addison Wesley Publishing

Newman, M. E. J. (2003). "The structure and function of complex networks." SIAM Review **45**: 167-256.

Nowak, M. A. (2005). The Pickup and Delivery Problem with Split Loads. Industrial and Systems Engineering, Georgia Institute of Technology. **PHD:** 131.

Perrow, C. E. (1999). Normal Accidents, Princeton University Press.

Pisinger, D. and S. Røpke (2007). "A general heuristic for vehicle routing problems." Computers & Operations Research **34**: 2403 – 2435.

Planco (2007). Economical and Ecological Comparison of Transport Modes: Road, Railways, Inland Waterways. Essen**:** 43.

Port of Rotterdam Authority. (2008). "Groei haven in gevaar door congestie." Perberichten, 2008 Retrieved 2009-08-14, 2009, from http://www.portofrotterdam.com/nl/nieuws/persberichten/2008/20080121_02.jsp.

Port of Rotterdam Authority (2008). Modal Split Containers 2003-2007. R. P. Authority. 2008, Rotterdam Port Authority**:** 1.

Potvin, J.-Y. and J.-M. Rousseau (1993). "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows." European Journal of Operational Research **66**: 331-340.

Press, W. H., S. A. Teukolsky, et al. (2002). Numerical Recipes in C++, the Art of Scientific Computing. Cambrigde, Cambridge University Press.

Rakt, B. J. A. v. d. (2002). Voorontwerp Afhandeling Wegcontainers in knooppunt Maasvlakten. Delft, TU-Delft TBM**:** 152.

Rakt, B. J. A. v. d., M. Melis, et al. (2005). Documentatie Simulatie Model Barge Handling. Rotterdam, INITI8**:** 30.

Røpke, S. and D. Pisinger (2006). "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows." Transportation Science **40**(4): 455-472

Røpke, S. and D. Pisinger (2006). "A unified heuristic for a large class of Vehicle Routing Problems with Backhauls." European Journal of Operational Research **171**: 750–775.

Ropke, S. and D. Pissinger (2005). "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows."

Roscam Abbing, R. M. (1999). Hoe Spoort het Water? Ontspoorde besluitvorming Betuwelijn; een analyse van genegeerde alternatieven. Delft, TUDelft/SISWO**:** 145.

Rotterdam Port Authority (2008). Container Throughput Hamburg - Le Havre Range Time Series Rotterdam Port Authority. R. P. Authority. Rotterdam, Rotterdam Port Authority.

Rotterdam Port Authority (2008). Modal Split Containers 2003-2007. R. P. Authority. 2008, Rotterdam Port Authority**:** 1.

RPA (2008). Container Throughput Hamburg - Le Havre Range Time Series Rotterdam Port Authority. R. P. Authority. Rotterdam, Rotterdam Port Authority.

Saleh, M. (2000). hard core of the System Dynamics research programme. The 18th International Conference of The System Dynamics Society. S. D. Society. Bergen, Norway, System Dynamics Society

Savelsbergh, M. W. P. and M. Sol (1995). "The General Pickup and Delivery Problem." TRANSPORTATION SCIENCE **29**(No. 1, February 1995): 17-29.

Schrimpf, G., J. Schneider, et al. (2000). "Record Breaking Optimization Results Using the Ruin and Recreate Principle." Journal of Computational Physics **159**: 139–171.

Shannon, R. E. (1975). System Simulation; the art and science, Prentice-Hall.

Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report. Glasgow, Scotland., Department of Computer Science, University of Strathclyde

Simon, H. (1957). Models of Man. New York, Wiley.

Smith, D. E., J. Frank, et al. (2000). "Bridging the Gap Between Planning and Scheduling." Knowledge Engineering Review **15**(1): 34.

Sommerville, I. (2006). Software Engineering, Pearson Education.

Stallings, W. (2001). TCP/IP and OSI. Business Data Communications. W. Stallings. Upper Sadle River, Prentive Hall**:** 91-96.

Sterman, J. D. (2000). Truth and beauty: validation and model testing. Business Dynamics: systems thinking and modeling for a complex world. J. D. Sterman**:** 845-891.

Strien, P. J. v. (1986). Praktijk als Wetenschap. Methodologie van het sociaal-wetenschappelijk handelen. . Asen/Maastricht, van Gorcum.

LITERATURE

Teleroute. (2009). "Over Teleroute."   Retrieved 2009-08-02, 2009, from
http://www.teleroute.nl/Holland/Nederlands/Home/About%20Us/page.aspx/1758?refererId=1
775.

Transport & Logistiek Nederland (2008). Geldende dieselprijs (af pomp in euro/ltr., excl. btw) per dag
vanaf 1 januari 2005 t/m heden, Transport & Logistiek Nederland.

Transumo (2009). VAN MAASVLAKTE NAAR ACHTERLAND. C. T. A. project. Rotterdam, Transumo.

van Hövell tot Westerflier, H. R. P. C. (2009). Smart Container Consolidation. Faculty Technology, Policy
and Management. Delft, Delft University of Technology. **MSc:** 120+.

Vangheluwe, H. and J. de Lara (2002). Meta-Models are models too. Winter Simulation Conference,
WSC. **1:** 597-605.

Velasco, N., P. Castagliola, et al. (2009). A Memetic Algorithm for a Pick-Up and Delivery Problem by
Helicopter Bio-inspired Algorithms for the Vehicle Routing Problem. Heidelberg, Springer
Verlag. **191**.

Vereniging de Binnenvaart. (2009). "Databank: De Binnenvaartschepen."   Retrieved 2009-05-19, 2009,
from http://www.debinnenvaart.nl/schepen_home/.

Verschuren, P. and H. Doorewaard (2007). Het ontwerpen van een onderzoek. Het ontwerpen van een
onderzoek. Den Haag, Uitgeverij LEMMA**:** 201-211.

Weerdt, M. d. (2003). Plan Merging in Multi-Agent Systems. Faculty of Electrical Engineering,
Mathematics and Computer Science. Delft, Delft University of Technology. **PHD:** 204.

Weise, T. (2009). Global Optimization Algorithms. **2009-06-26**.

Welters, P. D. H. W. H., D. P. d. Langen, et al. (2003). Port Economics I. Rotterdam, Erasmus
Universiteit.

Wieberneit, N. (2008). "Service network design for freight transportation: a review." OR Spectrum **30**:
36.

Wikipedia. (2009). "Simulated Annealing."   Retrieved 2009-08-22, 2009, from
http://en.wikipedia.org/wiki/Simulated_annealing.

Wikipedia. (2009). "Unit testing."   Retrieved 2009-06-22, 2009, from
http://en.wikipedia.org/wiki/Unit_testing.

Woltman Elpers, C. I. M. (2005). Parcel Tankers in the Port of Rotterdam, Research into the process of
handling parcel tankers in port and creating commitment for shortening the port time in a
multi-actor environment. TU-Delft: Faculty of Technology, Policy and Management. Delft, Delft
University of Technology. **MSc:** 70.

Zeigler, B. P. (2003). DEVS Today: Recent Advances in Discrete Event-Based Information Technology.
International Symposium on Modeling, Analysis, and Simulation of Computer Systems. I. C.
Society, IEEE Computer Society. **0:** 148.

Zijpp, N. J. v. d., M. C. J. Bliemer, et al. (1999). Microscopische Toedelingstechnieken voor de
Binnenvaart: Beschrijving Prototype Simulatiemodel. M. Bliemer, TU Delft.

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviated Name	Full Name
ALNS	Adaptive Large Neighborhood Search
APM-T	AP Moller Terminals
ARA-Ports	The ports of Amsterdam, Rotterdam and Antwerp
BC	Barge Captain
BCT	Bossche Container Terminal
BCTN	Binnenlandse Container Terminals Nederland
BO	Barge Operator
BOP	Bundled Operator Planning
CBRB	Centraal Bureau voor Rijn- en Binnenvaart
CH	Carrier Haulage
CTN	Container Terminal Nijmegen
CTT	Container Terminal Twente
DEVS	Discrete Event System Specification
DSOL	Distributed Simulation Object Language
DWH	Datawarehouse
ECT	Europe Container Terminals
ETL	Extract Transform and Load processes
FEU	Forty feet Equivalent Unit
GPS	Global Positioning System
HCSS	Hinterland Container Shipping System
HLH Range	Hamburg – Le Havre Range of Ports
ICL	Independent Container Lines
ITO	Inland Terminal Operator
KM	Kilometer
LNS	Large Neighborhood Search
MH	Merchant Haulage
MiCH	Merchant Induced Carrier Haulage
MinVW	Ministerie van Verkeer en Waterstaat

MSC	Mediterranean Shipping Company	
MTA	Meer-Terminal Afstemming	
PDP	Pickup and Delivery Problem	
PDPSL	Pickup and Delivery Problem with Split Loads	
PoC	Proof of Concept	
PoR	Port of Rotterdam	
PRA	Port of Rotterdam Authority	
PSA-HNN	Port of Singapore Authorities – Hesse Noord Natie	
QoS	Quality of Service	
RCT	Rotterdam Container Terminal	
RST	Rotterdam Shortsea Terminal	
SOP	Single Operator Planning	
TEU	Twenty feet Equivalent Unit, Container standard measure	
TLN	Transport en Logistiek Nederland	
TO	Terminal Operator	
TSP	Travelling Sales man Problem	
VITO	Vereniging van Inland Terminal Operators	
VRP	Vehicle Routing Problem	
WIT	Wansum Intermodal Terminal	

APPENDIX A.   ACTOR ANALYSIS

The appendix describes the actor analysis performed as a part of the system analysis. This describes the actors, their activities, goals and influence. The relation between the actors are also described. The analysis will show which actors are critical within the HCSS. The goal is draw a complete picture of the playing field. This actor analysis has been heavily based on the work by Van Andel (2007) and Van de Rakt (2002) and Connekt (2003).

ACTOR ROLES

An actor role consists of a number functions and has specific goals. Note that actor roles can be combined in one actor. The actor roles are described below:

Name	Client
Description	The client is the person who orders the transport to take place. A client usually owns the contents of the container, the container itself is usually owned by a carrier or a leasing company. It is important for the client that the cargo of the container reaches its destination within the set time and that the cargo is in proper condition. By ordering transport of cargo to its receiver the client provides the input for the system.
Activities	a. Orders the transport of containers to its receiver.
Interests	Getting its goods to its receiver.
Goals	a. Maximum profit. b. Maximum customer satisfaction. c. Minimal transportation costs. d. Maximal reliability transport.
Problem	Congestion in the ports. High transport prices.
Influence	Limited, a client can chose a different shipper. However clients tend to become more and more involved in the organization of transport; in such a context the influence of the client can be significant.
Resource	a. Production facilities b. Contractual relations with shippers and carriers.
Replaceable	Medium
Dependent	Medium
Critical	No

Name	Shipper
Description	The shipper organizes the transport of the container and takes care of the paperwork and other formalities which come with the transport. The shipper plans the transport, gives transport orders to the transporting parties and keeps the client posted on the progress of the transport. It is vital for the shipper that demands of the client are upheld and that the shipper's costs are minimal. The shipper has intimate knowledge of the system and adds value because of lack of transparency towards the client.
Activities	a. Arranges sea-transport with a shipping agent. b. Arranges hinterland transport with a barge-operator. c. Arranges last-miles transport with a transporter local to the inland terminal. d. Arranges transshipment of the transport at the hinterland terminal with the inland terminal operator.
Interests	Efficient and reliable transportation of the freight belonging to its customer

Goals	a. Maximal profit.
	b. Minimal transportation costs (price per container).
	c. Maximal QOS for customer – minimal lateness.

Problem	Congestion makes transportation more expensive and less reliable. High(er) transportation costs.

Influence	Shippers can improve the communication in the chain. The shipper can change the transport operators it is using (both barge operators and carriers).

Resource	a. Knowledge supply chain.
	b. Contracts with transport operators.

Replaceable	Low

Dependent	High

Critical	Yes

Name	Shipping Agent

Description	The shipping agent represents the carrier in the seaport. The agent takes care of the acquisition of new cargo for the carrier, and also controls the handling of the carriers' ships in the seaport. The agent is extremely dependent on the carrier, and the agent is evaluated by the carrier on the number of acquired transports and the efficiency of the handling.
	Although a shipping agent is the formal representative of a carrier in a port, this role can be executed by the carrier itself. A shipping agent is valuable to carrier because of its intimate local knowledge. There is a tendency among carriers to integrate such knowledge into their organization.

Activities	a. Forwards the transport assignment to the deep-sea carrier.
	b. Notifies the stevedore of the arrival of carrier ships.
	c. Assigns the stevedore with a transshipment assignment of containers from deep-sea vessels to connecting transport.

Interests	Good representation of carriers interests in port.

Goals	Minimal port time carrier ships.
	Maximal utility rate carrier ships.

Problem	Long port time ships.

Influence	Improve communication

Resource	Intricate knowledge of the port and actors,

Replaceable	Medium

Dependent	Low

Critical	No

Name	Barge Operator

Description	The barge operator manages a fleet of barges. These barges are either hired or owned by the barge operator. The barge operator gets transport orders from a shipper. These transports are usually between a terminal in the hinterland and a terminal in a deep-sea port. An operator has frequent services between the hinterland and the deep-sea port. Transports are assigned to these services. The service level of the operator and the terminals served by the operator determine the choice for a barge operator by a shipper.
Activities	a. Plans and assigns a transport to one of his barge captains. b. Notifies the stevedore of the arrival of a barge
Interests	Reliable and efficient transport of containers between seaport and hinterland terminals.
Goals	a. Minimal costs per container b. Minimal container exceeding QOS.
Problem	a. Little opportunities for growth due to competition, congestion and crisis. b. Decreasing margins.
Influence	a. Change of planning methods. b. Use different modalities. c. Horizontal cooperation . d. Changes in barge capacity.
Resource	a. Barges b. Customers c. Market knowledge
Replaceable	Problem Owner
Dependent	Problem Owner
Critical	Problem Owner

Name	Carrier
Description	The carrier takes care of the deep-sea transport of a container. He owns the ships that take care of the deep-sea transport and often owns the transported containers. The carrier gets his transport orders from the shipper by means of the agent. The choice for a carrier is made based on the service quality and its flexibility.
Activities	a. Plans the transport assignments on one of his vessels. b. Moves the container from one deep-sea port to another deep-sea port. c. Notifies the shipping agent of the arrival of ships. d. Containers (and their documents) flow from the carrier to the stevedore.
Interests	Reliable intercontinental transportation of containers.
Goals	1. Maximal profit. 2. Minimal transportation costs. 3. Minimal turrnarround in ports.
Problem	a. Economic crisis. b. Congestion in ports worsens the turnaround of vessels in port.
Influence	a. Changing of ports b. Contractual relationships with terminals.
Resource	a. Deepsea vessels

	b. Customer base
Replaceable	Low
Dependent	Low
Critical	No

Name	Stevedore
Description	The stevedore (Terminal Operator - TO) transships the containers from deep-sea vessels to connecting transport, the stevedore always provides a temporary storage for containers. The location used for transshipment is the terminal. The stevedore gets his transshipment orders from the agent. The stevedore has a contractual relation with the agent (and the carrier) and his main goal is satisfy the needs of these actors.
Activities	a. Transships the containers from a deep-sea vessel to its connecting transport. b. Temporarily stores containers at its terminal.
Interests	Transshipping and storing containers.
Goals	a. Efficient terminal operation. b. Meeting contract regulations with carriers.
Problem	a. Current economic crisis. b. Congestion in ports.
Influence	a. Change the demands for treatment b. Change capacity
Resource	Quays, cranes and personnel.
Replaceable	Low
Dependent	High
Critical	Yes

Name	Customs
Description	Customs is a government agency that is responsible for enforcing import/export regulations. The customs perform checks on all incoming and outgoing containers. Without customs clearance a container cannot be transported from a deep-sea terminal.
Activities	a. Checks the contents of container. Not all containers are check but a sample is taken, the sample size is based on risk profiles for types of containers. b. Clears containers for import/export.
Interests	Customs check and uphold Dutch and European laws concerning the import and export of goods.
Goals	a. Efficient checking of containers. b. High quality checking process.
Problem	Increase in container volumes makes checking of containers harder.

Influence	a. Different checking processes. b. Legal powers.
Resource	Laws and regulations on import and export.
Replaceable	Low
Dependent	Yes
Critical	No

Name	Barge Captain
Description	The Barge Captain (BC) is responsible for the container transport in the hinterland. He gets his transport order from the barge operator. The barge captain must follow rules and regulation for dangerous goods, stowage, resting times, and waterway usage.
Activities	a. Communicates his sailing with waterways authorities. b. Communicates his dangerous payload with waterway authorities. c. Moves containers to its destination. d. Load/Unloads containers at terminal. e. Communicates stowage plan with stevedore. f. Communicates stowage plan with inland terminal operator.
Interests	Reliable and durable transportation of containers by barge.
Goals	a. Keeping to contract regulations b. Optimal transportation process c. No accidents.
Problem	a. Less work because of economic crisis. b. Congestion in ports. c. Unreliable plans.
Influence	Ad-hoc planning in order to deal with congestion and unreliable planning.
Resource	a. Barge. b. Other captains.
Replaceable	Medium/High
Dependent	Medium
Critical	Yes

Name	Inland Terminal Operator
Description	The Inland Terminal Operator (ITO) transships the containers from barges to its connecting, last-mile, transport. The ITO always provides a temporary storage for container and is the hinterland counterpart of the stevedore. The inland operator gets transshipment orders from the shipper. An inland operator is usually chosen based on its location in the hinterland.
Activities	a. Transships the containers from a barge to its connection transport. b. Stores containers.

Interests	Efficient transshipment and storage of containers.
Goals	a. Maximal profit. b. Large client network. c. Reliable connections with seaport terminals.
Problem	a. Limited growth. b. Bad connections to seaport due to congestion
Influence	Limited.
Resource	a. Terminals b. Own fleet of barges and trucks (sometimes)
Replaceable	Low
Dependent	High
Critical	Yes

Name	Local Transporter
Description	The local transporter is usually a trucking company that takes care of the transport between inland terminal operator and receiver. The Local Transport gets its orders from the shipper and is usually chosen based on its location.
Activities	a. Transports the containers between an Inland Terminal Operator and a Receiver.
Interests	Reliable and durable transportation of container by truck.
Goals	Maximal profit.
Problem	-
Influence	-
Resource	Truck
Replaceable	High
Dependent	Low
Critical	No

Name	Receiver
Description	The receiver is the endpoint of the containers.
Activities	a. Receives the containers from the local transport.
Interests	The reception of goods in order to conduct business.
Goals	a. Timely reception of goods. b. Adequate QOS for the delivery of goods. c. Minimal costs for the delivery of goods.
Problem	Goods are delivered late, at high costs.

Influence	Influence on the client.
Resource	-
Replaceable	Low
Dependent	Low
Critical	No

Name	Barge Owner
Description	The Barge Owner manages one or more barges and is responsible for the management of these resources.
Activities	a. Rents barges to barge operators. b. Maintains barges.
Interests	The effective management of its fleet of barges.
Goals	a. Keeping to contract regulations. b. Optimal use fleet.
Problem	a. Low margins in market. b. Competition from other owners. c. Threatening initiatives such as bundling.
Influence	Negotiating contract terms.
Resource	a. Barges. b. Contracts with barge operators.
Replaceable	Low/Medium
Dependent	High
Critical	Yes

CARRIER VS. MERCHANT HAULAGE

The shipper controls the process, by deciding who transports the cargo, how the transport takes place and when it takes place. The existence of a shipper is mainly due to the lack of transparency within the container market, as transparency increases the added value of a shipper decreases. The latter means that other parties can take over the role of shipper. The role of the shipper can be integrated with other roles, this changes the form of the transport arrangements (Rakt 2002; Transumo 2009):

1. Merchant Haulage (MH). Merchant Haulage is transport arranged by a client or a shipper. The advantage of MH is that the organizer has more control over the transport process. The disadvantage is number of arrangement that must be made. The size of MH cargo is usually smaller that CH cargo, this means MH transports are less significant for the transporting parties. This makes it harder for the client/shipper to negotiate about prizes (volume discounts) and quality of transport service.
2. Carrier Haulage (CH). Carrier Haulage is transport arranged by the carrier or its shipping agent. The advantage for the client is that the carrier is a one-stop-shop for its transport, the advantage for the carrier is that it has more control over its process and the calls made by its vessels. The carries in fact consolidates its cargo and is able to optimize its process. For other parties CH can have downsides, the client will have less influence on the process, and because of the size of CH

transport other transporting parties (Stevedores, Barge Operators, Inland Terminal Operators, etc) are more inclined to bargain with a carrier.

3. Merchant induced Carrier Haulage (MiCH). In this case clients are capable of offering large cargo sizes to carriers. Carriers will then take care of the transport arrangements. The client remains in control; it can choose which carrier to use, while the burden of arranging the transport lies with the carrier. This is only attractive to carriers when the volume of the containers offered is large enough.

OTHER ACTORS

Outside of the primary process there are other actors who are involved indirectly. These are mostly governmental, consulting and branch parties. These parties influence the process by setting the rules, creating incentives for change, providing better process information and lobbying. Governmental parties in this perspective often have conflicting interests, for example a national government wants to increase the transport by barge while some municipalities do not want the added traffic generated by a local inland terminal.

ACTOR RELATIONSHIPS

The relations between actors can be categorized as such:

a.   Control, one actor controls the other (arrows with fixed line).
b.   Influence, one actor can influence the other (arrows with dashed line).
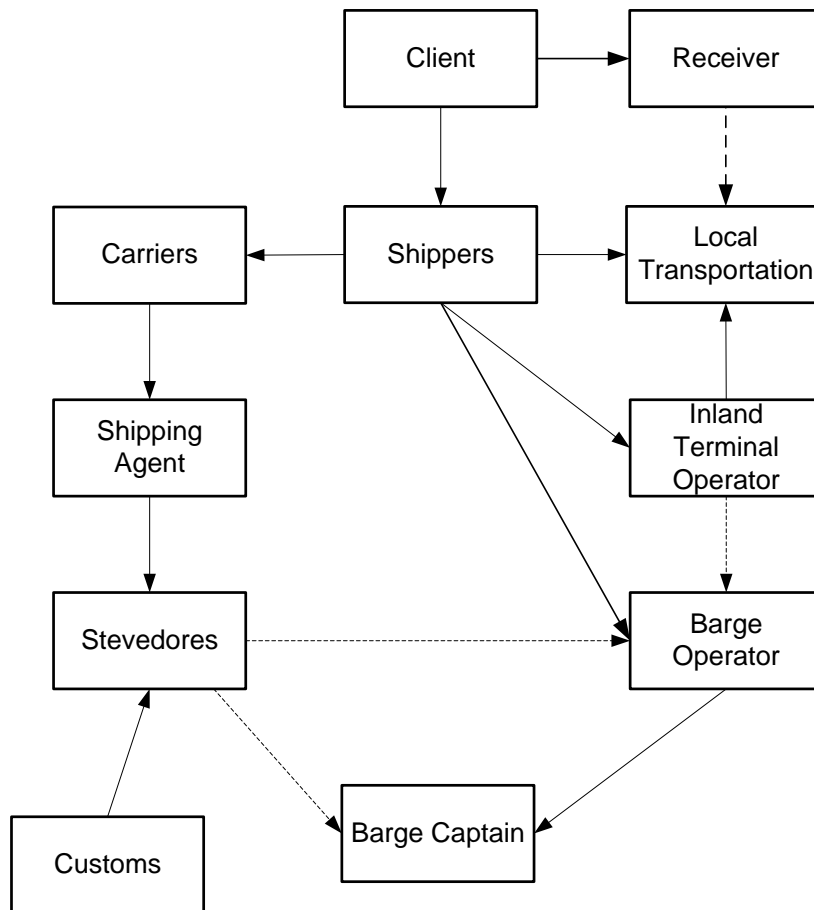
Figure 51 Actor Network

Figure 51 clearly shows the lack of relation between the barge operator and the stevedore.

APPENDIX B.   HCSS COST MODEL

This appendix describes the cost model. The goal of the cost model is to provide a clear and transparent way of estimating the cost generated by the HCSS. For commercial entities costs are the most important indicators in assessing their performance.

REQUIREMENTS

The requirements for the cost model are:

1. The cost model must be to approximate the costs generated by the barge operator's transportation processes. The actual cost structure of a barge operator depends on the scale of its operations and the way the operator is organized. This level of detail is not currently required, nor desired because of the sensitive nature of cost information. In order to keep the cost model simple and its application generic only the main cost elements need to be identified and used in for cost calculation;
2. Costs must be analyzable by a number of axes:
    o   Time, cost development over time;
    o   Barge, costs per barge;
    o   Route, costs per route;
    o   Region, costs per region.
3. The cost model must be able to determine the costs at lowest cost generating level. This creates transparency in cost calculation and allows for analysis of costs by the required axes;
4. All costs elements and their contribution to the cost function must be identified. This improves the transparency of the cost model.

ASSUMPTIONS

A model is a view on reality, it is a simplification and is biased. The cost model is no exception, this section shows the assumptions made in the model. The rationale behind many of these choices is that the cost model tries to capture the costs and revenues of the primary transportation process and is not an exhaustive model for barge operator costs. This leads to the following assumptions:

1. The staff costs are not taken into account. These are the costs made during the booking and planning processes. The costs for these processes are dependent on the workload of the organization and is dependent on the demand for transport and the complexity of these transportation demands. The staff processes are not taken into because they are not a part of the primary transportation process and the lack of information on their cost structure.
2. Terminal handling and storage costs are not taken into account. These are the costs of the terminal operator. These processes are partially outside of the demarcated system. BCTN and VITO operate the inland terminals themselves and the costs for handling at the inland terminal are theirs. The costs are not taken into account because of the focus on barge operations;.
3. Inflation is not taken into account, because this would introduce time dependencies for each elements' weight. This would also compromise the comparability of transport costs and revenues over time. It is assumed that the cost elements weights do not change and that their current values are representative.
4. Port charges are not taken into account. This is a small cost factor and are only applied when visiting a seaport. The way of calculating port charges is different for each port. The port charges are not taken into account because of their minor impact on the total costs and the complexity added by their calculations.
5. The costs for lock and waterway links usage are not taken into account. They are not taken into account because of their minor impact on the total costs.
6. The distance a barge travels between two terminals is always the shortest path between these terminals. There is only limited information available on the paths taken between calls (gps data is only available since april 2007 for a limited set of barges), so all barges are assumed to take the shortest path. This assumption is validated by analyzing the available GPS trails.
7. The penalty costs for delivering low Quality of Service is equal to that of transporting the container by truck. The (potential) missing of closing time is the most common form of deviation from QOS, this is usually solved by transporting the container by truck.
8. The distances between terminals using road or using water infrastructure are roughly the same. For a distances between a terminal pair by road, the distance by water is used. This because of a lack of time and the small error it causes.
9. The fuel consumption of a barge is only dependent on barge size. The draft and the speed of the barge are not taken into account. Relating the draft and speed of the barge to the fuel

consumption required information on the engine and hydrodynamic properties of the barge. These are not readily available.

10. The rent for a barge is only dependent on its size. It is assumed that barges of the same size have the same costs.

COST FUNCTION

There are two major factors which drive the costs: barge activities and transports executed by truck. Both are the cost drivers on the lowest level and both are the direct result of the planning process. The costs for transports trucked are quite simple, they are dependent of the performance required for trucking the transport and the costs per truck per kilometer.

(1) *costs = trucked_transport_costs + barge_activity_costs*

(2) *trucked_transport_costs = transport_performance ** **truck_cost_performance**

The costs for barge activities are a bit more complicated because there are two different types of activity: calls and hops. However from a costs perspective all activity types have the same cost elements: barge time costs and travel costs. The barge time is the time that the barge is used for operations (and is the sum of call and hop durations for a given period) by a barge operator, the barge operator rents a barge for a price. This price is dependent on the size of the barge and the availability of the barge (some barges are only active during weekdays). The travelled distance costs are specific to hop activities (there is no movement during call activities). The travel costs are dependent on the fuel consumption of the barge, the fuel prices and the distance covered by the barge. The fuel consumption is dependent on the barge size. The fuel price fluctuates heavily over time and is modeled as such.

(3) *barge_activity_costs = barge_time_costs + barge_travel_costs*

(4) *barge_time_costs =* **barge_rent** * *barge_activity_time*

(5) *barge_travel_costs =* **barge_fuel_consumption** * **fuel_price** * *barge_activity_distance*

The barge rent, barge fuel consumption and truck cost performance parameters have been estimated using a report written for one the BCTN terminal, CTT (2008). The truck cost performance is a single value which could be retrieved from the report. Although the report contained some data on barge rent and barge fuel consumption and their relationship to barge size, there was complete data set. This data has been estimated using a two power functions, one estimating the relation of barge size with the barge rent per TEU and the other estimation the relation between barge size and fuel consumption per TEU. The fuel prices were not estimated and could retrieved by getting the prices for red diesel fuel from the website of Transport en Logistiek Nederland (TLN).

VERIFICATION

The structure of the cost model has to conform with the conceptualized system and has to meet the set requirements. This was done using a structural analysis. This yielded that trucked transports had to be added to the cost model. The cost model has also been checked for dimensional consistency, all dimensions matched. The results of the model have been verified using extreme value analysis and the model produced the proper results.

APPENDIX C.   HCSS SIMULATION MODEL

The appendix describes the objects in the simulation model and some of the important interactions.

MAIN OBJECTS

This section describes the main objects used in the simulation model. Note that these objects are a mix of passive domain objects which represent plans to execute and the active simulation objects.
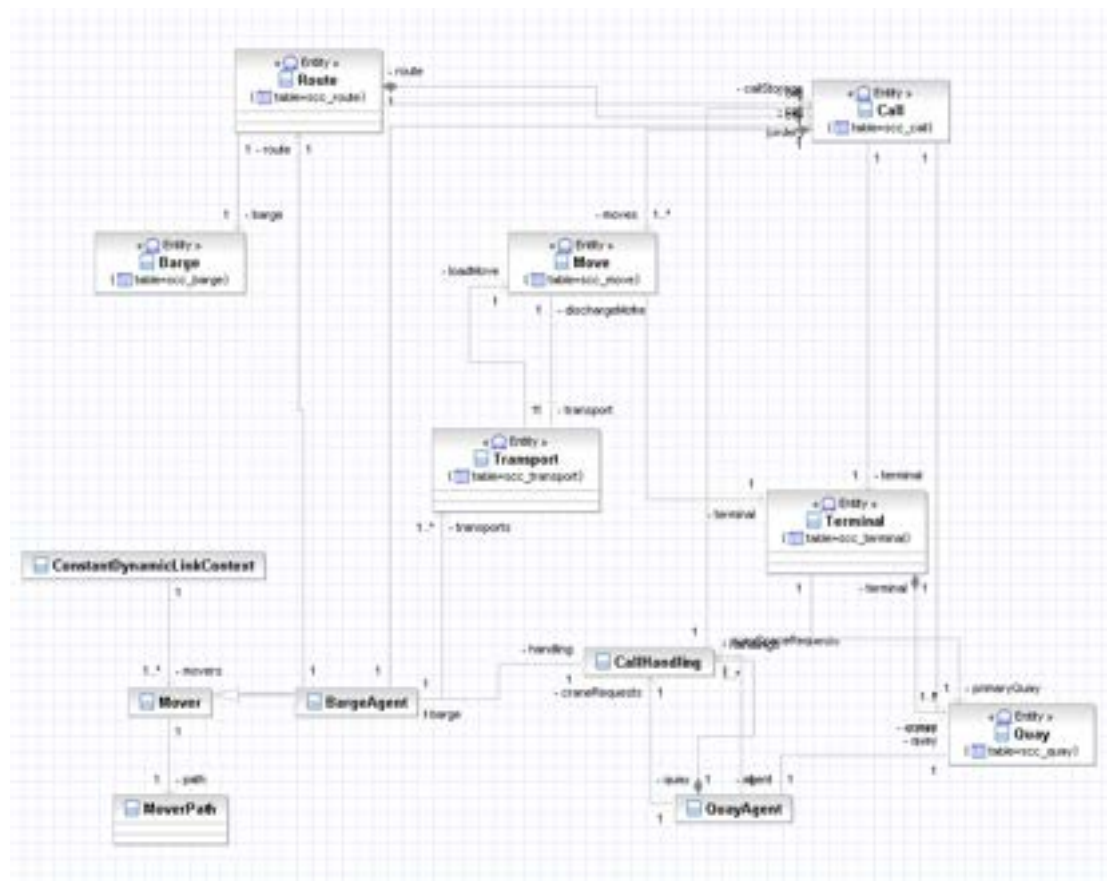
Figure 52 UML Diagram of Simulation Model classes

Figure 52 shows a UML class diagram of all classes relevant to the simulation model and their relations. Table 12 gives a short description of all relevant classes used in the simulation model.

Table 12 Description of Simulation Model classes

Package	Class	Description
nl.initi8.scc.domain	Barge	Planning object representing a barge. It contains all the relevant properties of a barge including methods for costs calculations and a speed distribution.
nl.initi8.scc.domain	Call	Planning object representing a call executed by a barge at a terminal and a quay. The Call defines which moves are executed. The call class is one of the core classes in planning. Because of this the call has extensive methods for manipulation and cost calculation.
nl.initi8.scc.domain	Move	Planning object representing the move of a container onto or from a barge during a given call. The move object ties a transport to a call.

nl.initi8.scc.domain	Quay	The quay is the location at which a call is executed. A quay has a capacity and can be limited in terms of length or draft. A quay always belongs to a terminal.
nl.initi8.scc.domain	Route	The route describes a series of consecutive calls for a barge. This is one of the core classes used for planning, that class has extensive methods for manipulation and cost calculation in order to accommodate this.
nl.initi8.scc.domain	Terminal	The terminal is geographical location at which a call is executed. The actual call is executed at a quay.
nl.initi8.scc.domain	Transport	The transport is the planning objects that represents the container that actually has to be move from its origin to its destination. The transport has a load and discharge move associated with it.
nl.initi8.scc.simulation.model	BargeAgent	The barge agent is a simulated object that executes a number of calls assigned to it. The barge agent is the extension of the mover and it uses the mover code to move from terminal to terminal.
nl.initi8.scc.simulation.model	CallHandling	A call handling is a messaging object between the QuayAgent and the BargeAgent used for the execution of a call. By moving the interactions to a dedicated object the management of both quay ad barge agent state became much more straight forward.
nl.initi8.scc.simulation.model	ConstantDynamicLinkContext	This context object is attached to a link and determines if a mover can enter the link and keeps track of activities on a link. The move mainly interacts with this class.
nl.initi8.scc.simulation.model	Mover	The mover class is an abstract class which represents an object which can move through the (waterway) network from an origin to a destination.
nl.initi8.scc.simulation.model	MoverPath	The mover path is the path taken by a mover during its travel process.
nl.initi8.scc.simulation.model	QuayAgent	The QuayAgent is a simulated object that takes care of the quays end of the call handling process. In order to do this it manages by crane and quay resource, it transships containers and it communicates with the barge agent.

INTERACTIONS

This subsection describes the most important interactions of simulated objects.

Calling Process

The execution of calls by a barge agent is described in Figure 53. Note that this is a conceptual description of the calling process – the implementation is somewhat different.
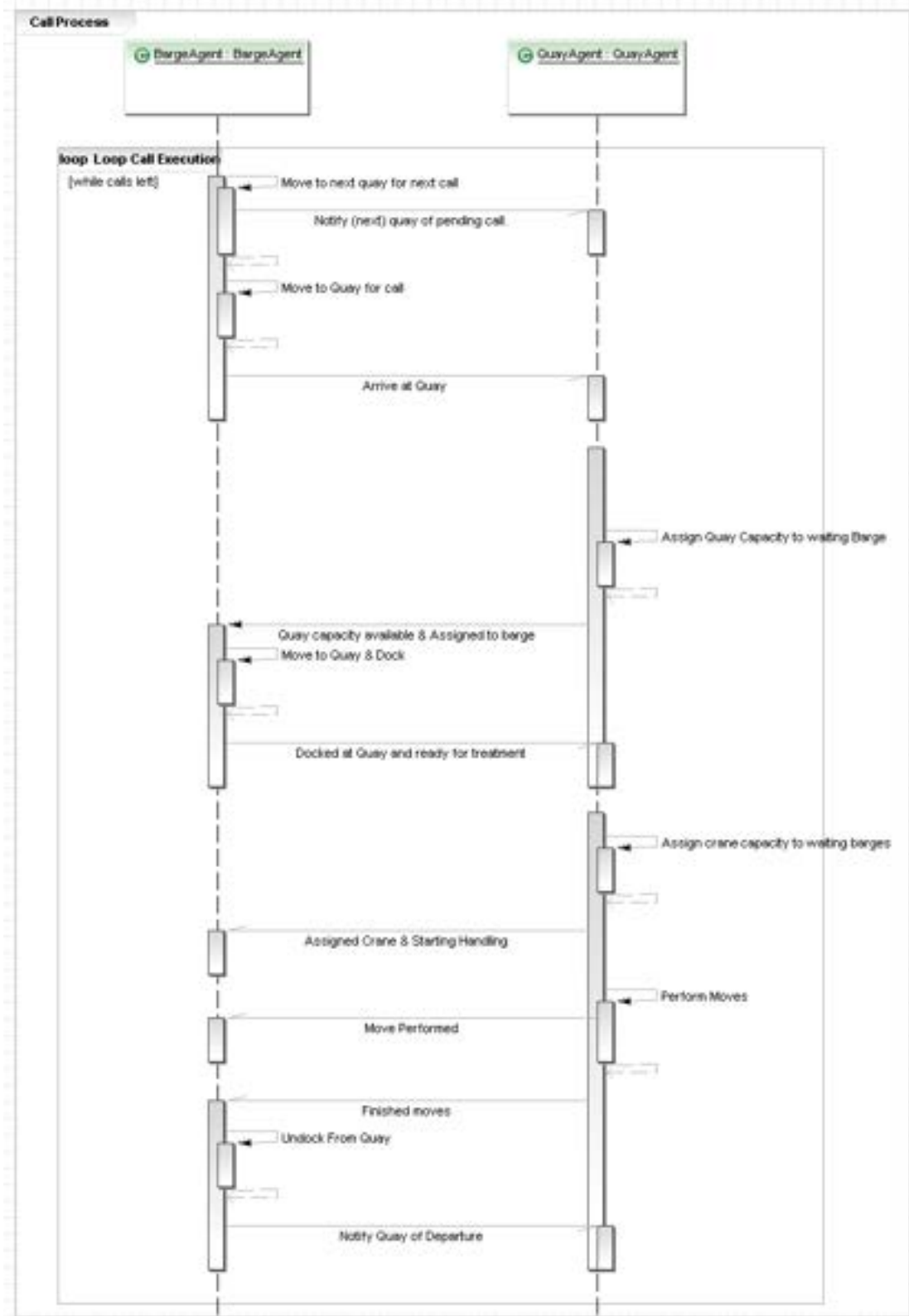
Figure 53 UML Interaction Diagram Calling Process

Mover Travel Process

Figure 54 shows the travel process of the mover. Note that this is a conceptual description of the travel process – the implementation is somewhat different.
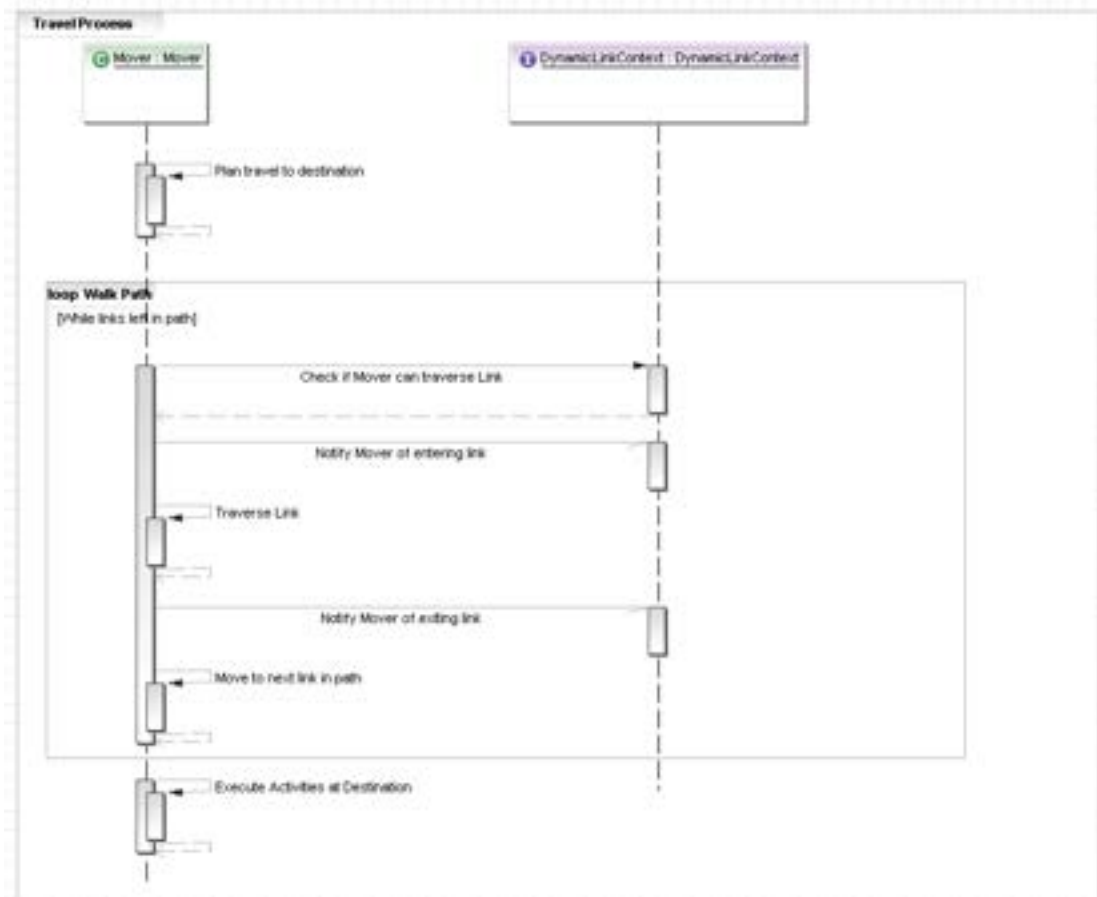
Figure 54 UML Interaction Diagram Mover Travel Process

APPENDIX D.   BARGE OPERATOR DATAWAREHOUSE

This appendix describes the datawarehouse constructed in order to analyze performance of Barge Operators.

REQUIREMENTS

This section describes the requirements for the barge operator datawarehouse. These have been based on personal communications with BCTN and other barge operators, the experience of INITI8 with the analysis of logistic systems and common best practices.

| Id	Name	Description
1	Usability	The data warehouse must be usable. It must be able to contain the core processes of barge operators and be able to contain reference data needed for analysis. The datawarehouse needs to store data in the following categories: Container transport data. The data warehouse must be able to store all the data needed for the analysis of container transports. This is information on the origin and destination of the container, the client, the shipper, the deep-sea carrier involved, the time in which the container needed to be transported, the time is which the container was transported, the size of the container, the contents of the container, the transporting barge, the load call and the unload call. Call data. The datawarehouse must be able to store data on the call and execution of calls. This is information on the terminal which is being called at, the barge used for transportation, the containers loaded during the call, the containers discharged during the call, the planned calling interval and the executed calling interval and waiting times. Analysis data. The datawarehouse must be able to store (reference) data which is needed for analysis. Examples of this are regional data.
2	Accuracy	The data in the datawarehouse must be an accurate representation of both the current as historical situations. The data in datawarehouse does not need to a real time reflection of the processes of the barge operator, but should be at least recent. Data must be loaded in the datawarehouse within a week after its last modification. This implies that the data loading processes must be quicker.
3	Traceability	Data in the datawarehouse must be traceable in order to validate the data, find sources of the data and to audit the data. The datawarehouse will integrate different types of data coming from different sources; tracking of data becomes impossible when no source metadata is stored with the data.
4	ACID	The datawarehouse must support Atomicity – Consistency – Integrity – Durability. This makes sure that data is valid, is reachable and remains that way.
5	Accessibility	The datawarehouse must provide easy access to its containing data for different frontends. The data itself is important, but as long as it's unreachable it is useless. The environment chosen for storage must therefore support Open DataBase Connectivity (ODBC) and should support Java DataBase Connectivity (JDBC).
6	Integration	The datawarehouse (and its surrounding infrastructure) must be able to store data from a number of barge operators using the Modality system into one integral data collection. The problem is that there exists a great deal of overlap between some of the data (for example client data, barge data and terminal data). Overlapping items must be stored as one item.
7	Responsiveness	The datawarehouse must be able to quickly deliver the data required by its users. It should not take too long for regular requests to be executed. This requirement is almost impossible to quantify, it is entirely dependent on the user; a user should perceive the performance as adequate.
8	Scalability	The datawarehouse must be scalable. It must able to contain the data of multiple barge operators and be able to deal with multiple users without significant service degradations.

DATA SOURCES

The quality of analyses made using the datawarehouses is dependent on the data in the datawarehouse. The data in the datawarehouse comes from a multitude of sources. These data sources are discussed in this section.

MODALITY

The first and foremost source for data is the registration system BCTN uses. BCTN, like most barge operators, uses 'Modality' as its primary system for conducting operations. Modality is a transactional system specifically targeted at inland container shipping and allows (among other functionality) an operator to register the bookings and transports, plan barge voyages and send invoices to customers (Modality Software Solutions 2009). BCTN currently runs four separate instances of Modality, one for each terminal. For the development of the datawarehouse data dumps of the Modality systems of CTN, BCT and WIT were available.

Modality is a transactional system and is designed to be very efficient at that. This is a far cry from the requirements for a datawarehouse. In practice this means that there is an impedance mismatch between the data model used by modality and the data model used by a datawarehouse. A prime example of this is the lack of call registration in modality: modality stores load and discharge voyages in which appointments are stored for loading and discharging containers. A combination of load and discharge (which are part of different voyages) appointments at the same terminal at the same time is a call. Some data is not registered at all because it is not needed for the process, for example the length of the voyages and the execution time of the calls. This is covered by the additional data sources described in this chapter.

Modality uses an Oracle 8i database for the storage of data. The application itself is created in a proprietary environment build directly on top of the Oracle 8i environment. The Modality database consists of 67 tables, of which 11 are needed for filling the datawarehouse.

GPS DATA CARRIERWEB

Data on the execution of calls is not stored in Modality. Terminal operators either do not register such information or do not want to discern such information. A different source for reliable data on the execution calls is needed.

On over 40 barges equipment of CarrierWeb has been installed which allows barge operators to follow their barges (and even communicate with them). INITI8 monitored these barge and collected Global Positioning System data in order measure waiting times at ECT Delta's Hartelhaven facility. This data was collected for BCTN's barges and can be used to determine when calls were executed and what the duration of the call was.

The CarrierWeb equipment sends a GPS signal every minute to CarrierWeb's central server by means of UMTS. The data is transferred to INITI8's severs by means of the CarrierWebEAI service. This service synchronizes the databases of CarrierWeb and INTI8 every minute.

GOOGLE EARTH MAP

Some of the analyses require information on the execution of calls and the distances between terminals. This requires geographical information. This source of this geographical information is a map constructed in Google Earth. The map contains all elements relevant to BCTN's barge transportation processes: a network connecting all significant terminal location to each other and polygons describing container terminal service areas.

The advantages of using Google Earth as a mapping tool are the free and public availability of the application, the wide spread use of the application and the open standard, the Keyhole Markup Language (KML), in which Google maps are stored. Google Earth allows the user to add named geographical objects such as points (Placemarks), Polygons and Polylines. It also allows for the organization of these objects in a folder structure.

However Google Earth is not a true mapping tool that is capable of expressing the links between geographical objects, so network modeling cannot directly be achieved. This leaves implicit modeling of networks; this has been done by using a strict naming convention which contains the name of the object and its type. Unfortunately this also means that Google Earth cannot help in validating the network structure, this must be done during the map interpretation process. The major advantage of

Google Earth is the availability of, recent, aerial images on which locations and links can be easily plotted.

The current map (as of May 2009) represents most of the Dutch hinterland waterways and is extended to the port of Antwerp in Belgium and the Rhine ports in Germany (until Basel). It has over 20000 coordinates and over 1500 objects in it. The required accuracy of the map makes the map-making process sometimes very cumbersome and time consuming (especially in port area's).

DESIGN BARGE OPERATOR DATAWAREHOUSE
This section discusses the design of the barge operator data warehouse (BODWH). The design has been based on the requirements set, the available data and best-practices in DWH design. A DWH is actually a database which is targeted at analyzing data not registering it; as a consequence the design for a datawarehouse is different from a transactional database. The datawarehouse itself is useless without data. It needs to be loaded in the datawarehouse; this is done using Extract Transform and Load processes. Access to the data is required in order to analyze is. This access can be done using a frontend tool.
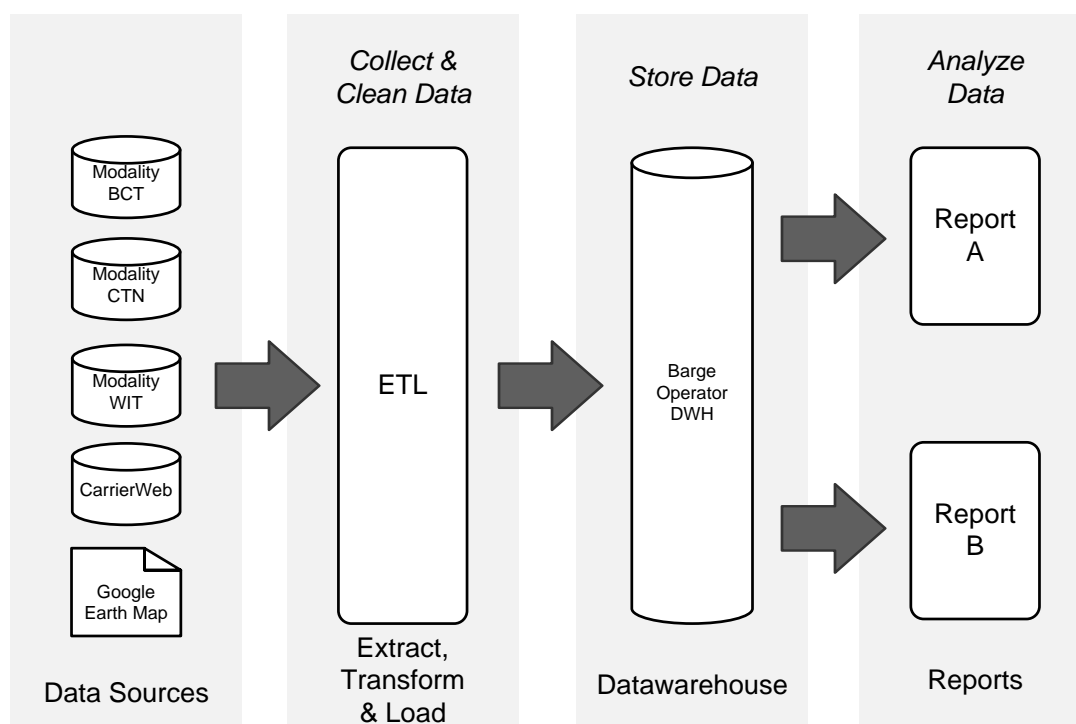
Figure 55 Conceptual Overview BO Datawarehouse.

Figure 55 shows the conceptual design of the datawarehouse, each gray column represents a different component and the arrows depict the flow of data. Over the course of this section these components are described:

a. Structure: The structure determines what data can be stored and how this data is related. The structure not only stores data on transports and barge activities but also supporting data.
b. ETL: The data to be stored in the data warehouse comes from many different sources. In some cases the data needs to be converted from their source representation to the data warehouses representation, in other cases the data is combined with data coming from other sources, the data conversion and combination is done during Extract Transform and Load (ETL) processes (Mundy, Thornthwaite et al. 2006).
c. Output: this describes how the data the data can be accessed, reported and analyzed.

This section is not a complete description of every aspect of the design; only the most important elements are described.

STRUCTURE BARGE OPERATOR DATAWAREHOUSE

The datawarehouse has been designed with the two main areas of concern in mind: container transport and barge activity data. The structure of the datawarehouse has been created using the dimensional approach (Mundy, Thornthwaite et al. 2006; Navathe and Elmasri 2006). The main reason for choosing this approach over other approaches (such as the normalized one) was the ease of use for the end-user and the generally better performance. In the dimensional approach data is partitioned by subject areas, which reflect a category of data, for example: finance data or personnel data. Per subject area one or more fact tables are used, these fact tables contain measures on these subject areas and these tables are linked to a number of tables, which contain reference information. These reference tables are called 'dimensions' and provide the context for the data stored in the fact tables. Dimensions can be reused in multiple fact tables but are often related to a single subject area.
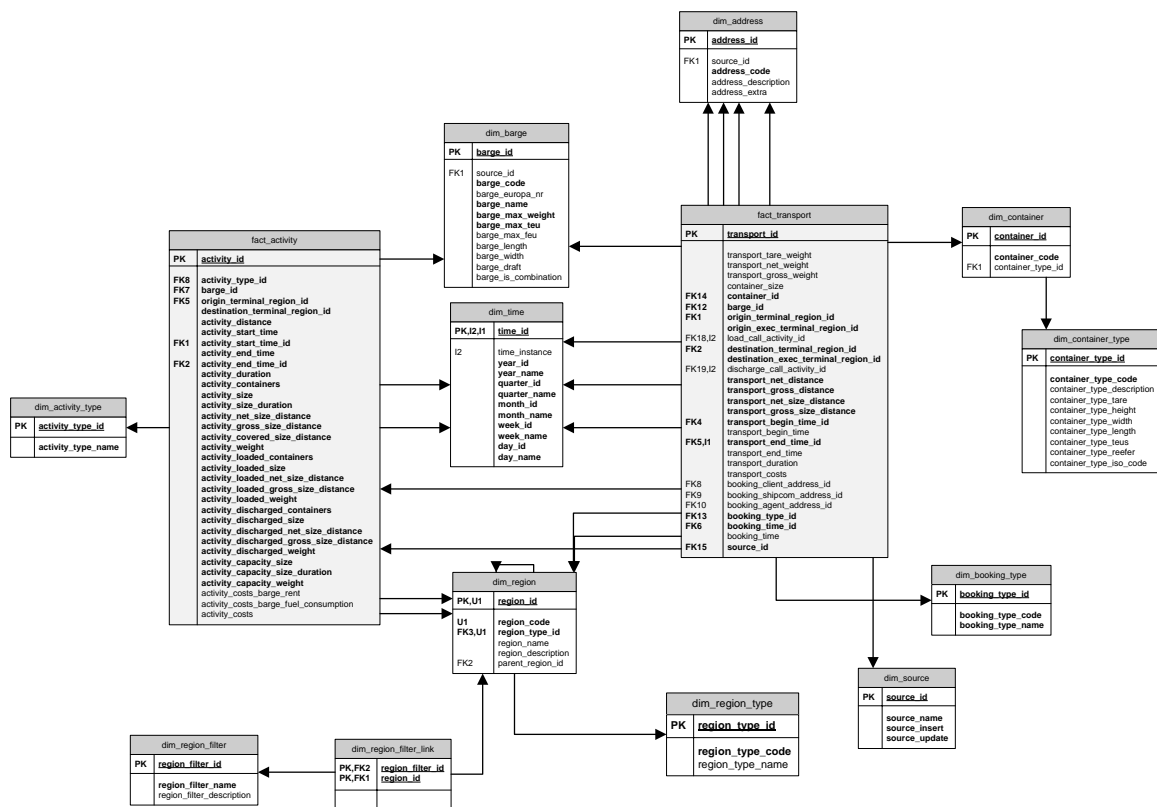
Figure 56 Small ERD of the Barge Operator Datawarehouse

Figure 56 shows the Entity Relationship Diagram (ERD) of the BO DWH the larger (grey) tables are the fact tables and the tables which surround the fact tables are its dimensions (this somewhat resembles a 'snowflake', hence the name snowflake-schema – which is commonplace in datawarehousing). The datawarehouse has created in Microsoft SQL Server 2005 Standard Edition. Table 13 describes the dimensions used, it shows and describes the type of dimensional data, which tables are used for storage and to which facts they relate.

Table 13 Dimensions used in the BO DWH

Reference Data	Tables	Description	Facts
Activity Type	dim_activity_type	The activity type is an ordinal dimension which describes the different types of activities possible (barge call and barge hop). The data in this dimension is determined by the number of activity types defined.	fact_activity

Address	dim_address	The address dimension describes the multiple actors which are involved in a container transport (client, shipper or carrier). The data in this dimension comes from Modality's address table.	fact_transport
Barge	dim_barge	The barge dimension describes the barges used for the transportation of containers. This dimension contains information on the properties of the barge: length, width, height, capacity, etc. The data in this dimension comes from Modality's barge table.	fact_activity fact_transport
Booking	dim_booking_type	The booking type dimension shows what the type of the transports booking is. This contains information on the priority of the transport and shows steps taken in order to complete transports. The data in this dimension comes from Modality's booking table.	fact_transport
Container	dim_container dim_container_type	The container dimension describes the actual container transported. There are two tables used: the container table and the container type table. The container table only contains the containers unique tracking number and a type. Whereas the type contains information on the properties of a container of the type given. The data in this dimension comes from Modality's unit and unit type table.	fact_transport
Region	dim_region dim_region_filter_link dim_region_filter	The region dimension describes the regions between which containers are transported. A region has a hierarchical notion to it, for example there is a terminal (ECT Delta) which is part of a port (Rotterdam). These is described using the same table by means of parent-child relationship (parent_region_id -> region_id), this allows for a flexible configuration of regions. Regions are also filterable. The data in this table is constructed using the Google Earth Map and Modality's address table.	fact_activity fact_transport
Source	dim_source	The source describes the modality instance a transport is coming from. This is to enable traceability of data. The data in this table is determined by the Modality instances used.	fact_transport
Time	dim_time	The time dimension is used to store time based properties: year, month, week, etc. Fact tables store data on events both using a key to the time dimension and using a complete date field. This has been done to allow for easy aggregations (using the time dimension) and exact time analysis (using the field) (Kimball 2004). The grain of this dimension is a day. The time dimension is	fact_activity fact_transport

generated in advance.

The datawarehouse has been designed around the two main areas of concern for the barge operator: container transports and barge activities. These fact tables use the dimensions described in Table 13, and also define a number of measures. Table 14 describes the fact tables.

Table 14 Description of the fact tables: container transports and barge activities

Name	Container Transports		
Description	This fact table describes the transport (to be) executed by a barge operator. Transports give information on the demand for transportation and also give information about the working and the performance of the barge operator. Transports that have been completed also have data on the barge that performed the transport, on the transportation times and on distances covered during transportation (gross distances).		
Table	fact_transport		
Dimensions	Address	The address dimension is used to describe the three (external) actors involved in the transport: the client who ordered the transport, the agent who is arranging the transport and the shipping company who is responsible for further transportation.	booking_client_address_id booking_shipcom_address_id booking_agent_address_id
	Barge	The barge dimension describes the barge used to transport the container from its origin to its destination.	barge_id
	Booking Type	The booking type determines what kind of booking the transport was a part of.	booking_type_id
	Container	The container describes which container was (/is to be) transported by the barge.	container_id
	Region	The regions describe the origin terminal and the destination terminal. However there can be a difference between the administrative terminal and terminal at which the container was loaded or discharged, for example container for the ECT Delta DNN terminals are often handled at the ECT Delta Hartelhaven terminal.	origin_terminal_region_id origin_exec_terminal_region_id destination_terminal_region_id destination_exec_terminal_region _id
	Source	The transport can be directly traced back to a Modality instance. The source dimension registers this instance. The source is also used for the identification of the barge operator to which the transport belongs.	source_id
	Time	The transport has numerous time fields associated with it: the booking time of the	booking_time_id transport_begin_time_id transport_end_time_id

		transport, the time at which the transportation started and the time at which the transportation ended.	
Measures	Weight	This measure describes the weight (KG) of the transport. The weigh can be divided into three weights: a net, a tare and a gross weight	transport_net_weight transport_tare_weight transport_gross_weight
	Size	The size of the container (TEU). This determines how much capacity is taken by the transport. This is dependent on the container type.	transport_teus
	Distance	The distance a transport will take (M). This measure can be divided into the minimum distance (net distance) a transport takes and the actual distance it took (gross distance).	transport_net_distance transport_gross_distance
	Performance	This measure expresses the performance required for transportation (TEU * KM) and is the combination of size and distance. This is useful for capacity calculations. The performance can be divided into minimum performance required and the actual performance delivered.	transport_net_size_distance transport_gross_size_distance
	Duration	This measure expresses the duration (seconds) of the transport.	transport_duration
	Costs	Costs of the transport (€).	transport_costs
Name	Barge Activities		
Description	This fact table describes the activities executed by barges. There are two types of activities: calls and hops. Both are stored in the same fact table, this means some data redundancy will exists, but this improves the ease with which data can be analyzed and gives a good overview of the planning of and the operations themselves.		
Table	fact_activity		
Dimensions	Activity Type	The activity type dimension describes the type of the activity (i.e. call or hop).	activity_type_id
	Barge	The barge dimension describes the barge that executed the activity.	barge_id
	Region	The region dimension describes the terminal at which the activity started (origin) and at which the activity ended (destination).	origin_terminal_region_id destination_terminal_region_id
	Time	The time dimension is used describe the beginning and end of the activity.	acitivity_start_time_id activity_end_time_id
Measures	Containers	This measure shows the containers on the barge after the activity, loaded during	activity_containers activity_loaded_containers activity_discharged_containers

	the activity and discharged during the activity.	
Performance	The measure shows the performance of the container loaded on the barge after the activity, the increase in performance due to loading, the decrease in performance due to unloading. The measures differ in net and gross performance. The contribution of the call to the performance is also measured.	activity_net_size_distance activity_gross_size_distance activity_covered_size_distance activity_loaded_net_size_distance activity_loaded_gross_size_distance activity_discharged_net_size_distance activity_discharged_gross_size_distance
Weight	This measure shows the weight (kg) of the containers on the barge after the activity, loaded during the activity, discharged during the activity and the maximum.	activity_weight activity_loaded_weight activity_discharged_weight activity_capacity_weight
Costs	This measure describes the costs (€) made during the activity. Both the total costs as its principal components are stored: rent and fuel.	activity_costs activity_costs _barge_rent activity_costs_fuel_consumption
Distance	The distance (meter) covered during the activity. This is 0 for a call.	activity_distance
Duration	The duration (seconds) of the activity.	activity_duration
Size Duration	This measure is a combination of the (maximum or loaded) size of the containers on the barge during the activity and the duration of the activity (TEU * second). This is used for the calculations of utility rates.	activity_size_duration activity_capacity_size_duration
Size	This measure shows the size of the containers (TEU) on the barge after the activity, loaded during the activity, discharged during the activity and the maximum.	activity_size activity_loaded_size activity_discharged_size activity_capacity_size

ETL PROCESSES BARGE OPERATOR DATAWAREHOUSE

As stated before the data for the data warehouse comes from a multitude of sources (Modality instances, a Google Earth Map and the Carrier GPS Data). This data is loaded into the data warehouse using ETL processes.
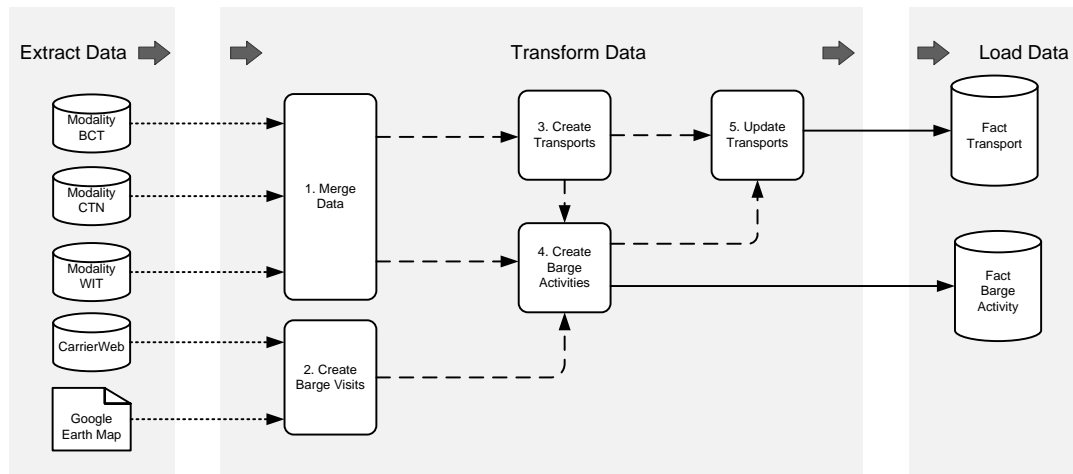
Figure 57 Overview of ETL processes

There are three sequential ETL steps:

1. Extract processes, these processes extract the required data from source systems. In order to do this step the location of the required data needs to be identified and a way of data extraction needs to be determined.
2. Transform processes, during this step the data is transformed from the source systems' representation to the target systems' representation. This step is by far the most involving, because all types of semantic mismatch between the systems have to be corrected during this stage.
3. Load, this step loads the transformed source data into the data warehouse.

The ETL process has mainly been implemented in Microsoft SQL Server Integration Services 2005 (SSIS). The advantage to SSIS is the large amount of ETL building blocks it provides and its visual designer (see Figure 58). The only exception to this is *Create Barge Visits* process, this required some very specific geographical processing which was not available in that version of SSIS; a customized transform was created in Java.
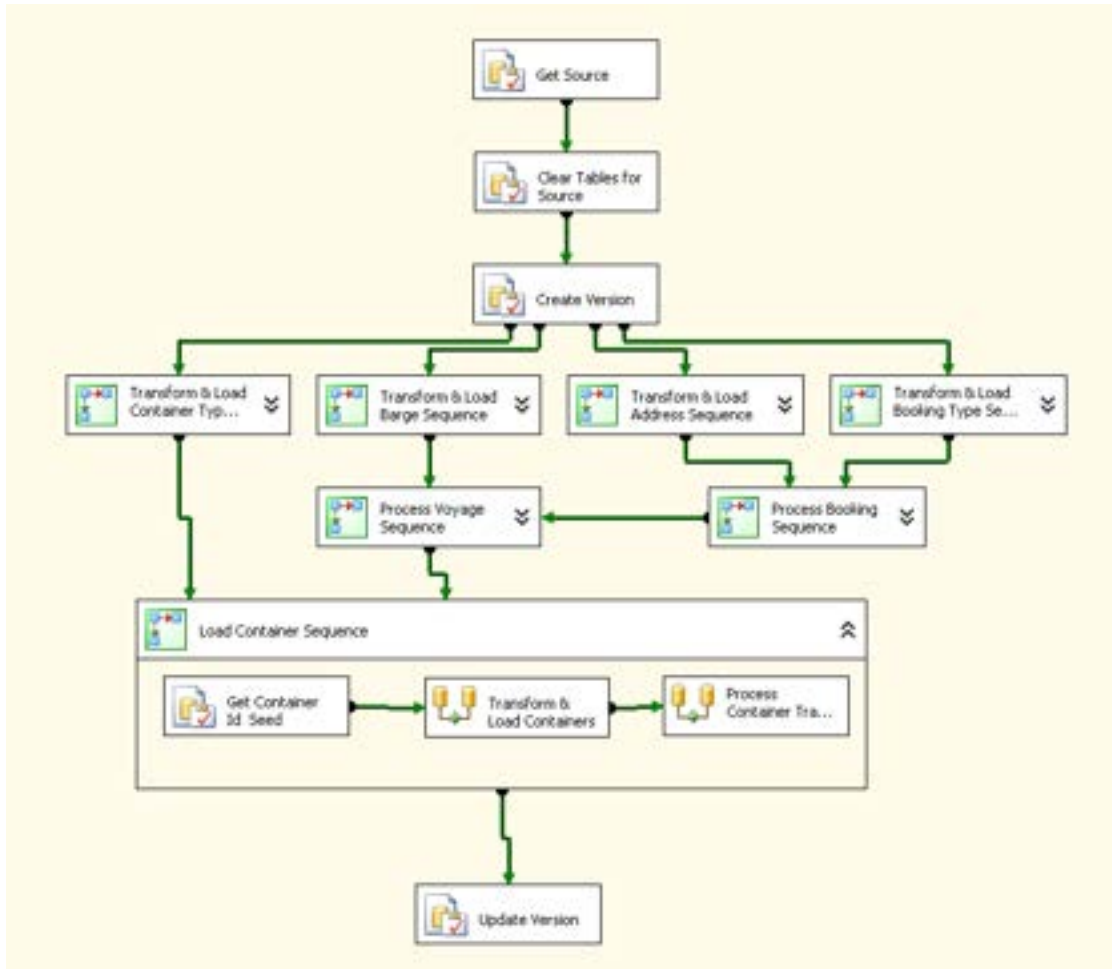
Figure 58 Screenshot of SSIS ETL package: Merge Data

Extract Processes

Data from three Modality instances, CarrierWeb and the Map had to be obtained:

1. Modality Instances. Data is loaded from 11 different tables in the database of a Modality instance in order to fill the datawarehouse with all the required data. This has been based on an initial analysis of the database (no documentation was available). The data from these 11 tables is extracted using SQL queries; the data is dumped into delimited text files. These files are then zipped and transferred using a secure FTP transaction to INITI8's servers. The data is then ready for the transformation processes.
    This extract method is adequate when the refresh rate of the data in DWH is relatively low (i.e. once a week), when a higher refresh rate is required a more elaborate extract process is required.
2. CarrierWeb data. CarrierWeb collects its data on its own servers. The aforementioned CarrierWebEAI service is used to transfer the GPS data from CarrierWeb's servers to INITI8 servers.
3. The Map. The data from the map is used as reference data. As a result it doesn't necessarily need to be loaded into the datawarehouse; it is needed during the transformation processes for the creation of barge visits. Because the map is used in the *Create Barge Visits* transformation the map had to be readable the Java platform. The map read using a java based KML reader and transformed into a usable map representation.

Transform & Load Processes

The subsection describes the transform and load processes. The transform processes make heavy use of a staging area; these are tables in which intermediate results of transforms and the initially extracted data is stored. After a successful run of the ETL process the staging area can be cleared (note that the

intermediate results were heavily used during testing in order to verify the results of the transformations). The results of the transformations are loaded in the datawarehouse. The tables below describe the transform and load processes.

Name	1. Merge Data
Description	This step is the workhorse of the ETL processes. It cleans and merges Modality data from the three instances. Dimension tables are loaded into the datawarehouse after this step. Fact tables require further processing, but are prepared during this transform. There is an overlap in the data contained by the three Modality systems used. For example over the course of time multiple terminals have used the same barge, this means that in each instance of Modality a reference to the same barge can exists. The same goes for other object types such as: clients, shippers, agents and terminals. In order to analyze the performance of transports and barge activities, these object types must be *canonicalized* (one normalized instance exists of each for the complete dataset). Because the naming of these objects is not consistent, the *canonicalization* process is a manual and very labor intensive job. In order to do this one of the instances of an object is designated as the *canonicalized* instance; the other instances of the object are then recoded to the *canonicalized* instance using a lookup table.
Transforms	(Modality)ADDRESS -> (DWH)dim_address
	(Modality)BARGE -> (DWH)dim_barge
	(Modality)UNITTYPE -> (DWH)dim_container_type
	(Modality)BOOKTYPE -> (DWH)dim_booking_type
	(Modality)UNIT -> (DWH)dim_container
	(Modality)BOOKING -> (STAGING)modality_booking
	(Modality)UNIT -> (STAGING)modality_transport
	(Modality)VOYAGE -> (STAGING)modality_voyage
	(Modality)VOYTERM -> (STAGING)modality_voyage_terminal
Steps	a. Transform & Load dim_address (Same steps Taken for b. dim_barge, c. dim_container_type & d. dim_booking_type) Read the dimension data from the file. Remove duplicates. Check if a canonicalized instance exists. If none exists write a new instance to the dimension table.
	e. Transform & Load dim_container Read container information from UNIT file. Assign dummy id to unknown containers. Lookup container type key Remove duplicate containers. Check if a canonicalized instance exists. If none exists write a new instance to the dim_container.
	f. Process voyages Read data from VOYAGE file per Modality instance. Assign source id for the Modality instance. Lookup barge & terminal data. Write voyage to modality_voyage.
	g. Process voyage terminal Read data from VOYTERM file per Modality instance. Merge with voyage data. Determine the execution data based on voyage and voyage terminal data. Resolve administrative and executive terminals Write voyage terminal visit to the modality_voyage_terminal table.
	h. Process bookings Read data from BOOKING file per Modality instance. Lookup keys for dim_booking_type and dim_address (for client, agent and shipping company). Write booking to modality_booking table.
	i. Process transports Read transport data from UNIT file. Split Import and Export transports (one UNIT record can contain both an import and an export transport

– this means two rows can be created).
Lookup the keys for modality_booking, dim_container and dim_address (for import, export and inland terminals).
Correct inland terminal address, when no valid terminal can be found using the source data.
Lookup voyage data by merging with modality_voyage_term and modality_voyage data
Assign dim_barge key based on voyage data
Set transport data based on voyage data.
Write transport to modality_transport.

Name	2. Create Barge Visits
Description	During this step barge GPS data is transformed into the visits of barge to terminals using the Google Earth map. This is the only ETL transform written in Java. This has been done using a batch processing framework called Spring Batch. The reason for using Java was that a spatial indexing technique was needed in order to match a GPS coordinate to a terminal or other location of interest such as a port. The indexing technique used is a PMR Quadtree (Hjaltason and Samet 2002), and the terminals and other regions in the map were loaded into the index in order to enable quick matching.
Transforms	(CarrierWeb)gps_point -> (Staging)barge_visit (MAP)location
Steps	Before processing starts: Load the map from the KML file. Create a PMR Quadtree based on the locations of interest in the map. Processing: GPS Coordinates are loaded per barge and are read in a time ordered way. The Quadtree index is queried using the GPS Coordinates; the index returns all locations the coordinate is a part of. If a visit for a location already exists, the visits statistics are updated. If no visit exists a new one is created for the location given. When a visit is not updates for a given time span (5 minutes) the visit is deemed to be finished and is written to the next step. Short visits are discarded as are visits with an extremely coordinate count. Visits are written to barge_visit table.

Name	3. Create Transports
Description	This transform creates the basic transport fact based on the staged transport. This transform is quite straight forward. The biggest work lies in merging bookings and transports into one fact and to lookup all relevant keys.
Transforms	(STAGING)modality_transport -> (DWH)fact_transport (STAGING)modality_booking
Steps	Read the transports from the modality_transports table. Apply a cut-off only the transports executed after 1-1-2006 are taken into account. Lookup booking data and merge it. Assign Executed Terminal to a transport based on voyage data. Translate modality keys to actual DWH keys for all dimensions. Set uninitialized columns to default values. Write transports to fact_transport.

Name	4. Create Barge Activities
Description	This transform process creates the activity fact. This is one of the more complicated transformations in the ETL process. This was mainly due to the fact that Modality does not store calls, it stores load or unload voyages which contain appointments, and these voyages had to be combined in order

	to obtain the correct activities. Another complicating factor was that incidentally BCTN barges make multiple stops at hinterland terminals, which results into multiple voyages which is actually the same voyage. The create barge activities transform does not use modality_voyage as its input, but uses the voyage information embedded in the fact_transport.
Transforms	(DWH)fact_transport -> (DWH)fact_activity (STAGING)barge_visit (Staging)fact_transport_activity
Steps	Read transports from fact_transport Correct the inland terminal of the transport in case of an administrative error. Split transport into a load move and a discharge move. A move contains a terminal, a barge, a (planned) execution date, a reference to the transport and a load/discharge flag. Sort the moves by barge and by (planned) execution date. Create calls and hops based on the sequence of moves. When subsequent moves have the same terminal and barge and are not too long apart (1 day) they belong to the same call, in other cases a new call activity and potentially a new hop activity is created. The moves are aggregated into transports-activity combinations and are written to a separate output. The activities are merges with the barge visit data. Note that this can actually result in LESS call and hop activities, because sometimes multi calls to the same region are merged into one call (the most common scenario are calls made to the ECT Delta Terminals these are often combined into one call at ECT Delta Hartelhaven). The activity distances for the (hop) activities are resolved. The activities are merged with the transport-activity combinations. Both the data for activities as transports are aggregated. The goal of this aggregation is to resolve information on the transports as the activities performance and other measures. The aggregated activity data is written to the fact_transport_activity. The activities require further processing. Keys for the activity fact are resolved. The activities are written to the fact_activity table.

Name	5. Update Transports
Description	This transform updates the transport fact with execution information: costs, gross distance and performance. This is the only step that has been created using a SQL query only.
Transforms	(DWH)fact_transport -> (DWH)fact_activity (Staging)fact_transport_activity
Steps	1. Read transports from fact_transport 2. Update the fact_transports with data in the fact_transport_activity table.

OUTPUT OF THE BO DWH

The goal of the datawarehouse is to measure the performance of Barge Operators. The data can be accessed in a couple of ways. First a simple data-cube has been created in SQLServer Analysis Server 2005, this sits on top of the DWH and supports natively supports the dimensional approach and allows the users to perform multi-dimensional analysis on the data. The major drawback of such a cube is that performing non standard (ad-hoc) analyses can be quite complex, as this requires intricate knowledge of the Multi-Dimensional eXpression (MDX) language. Another way of accessing the data is by writing accessing the data directly by means of SQL queries.

During this thesis the datawarehouse was mainly used to gain insight into the processes of BCTN, to create experiments and to verify and validate the results of experiments.

VERIFICATION & VALIDATION OF THE BO DWH
Sommerville (2006 chapter 22) states that the verification and validation builds confidence in the software created, this does not mean that the tool must be perfect but good enough for its intended use. Verification checks if the datawarehouse has been constructed too specification i.e.: *"are we building the product right?"* (Sommerville 2006 chapter 22). Whereas validation checks if the actual

result is usable i.e.: *"are we building the right product?"* (Sommerville 2006 chapter 22). This section tries to answer these questions.

The validation and verification have been executed using a number of checks and tests. The reason why validation and verification are in one section is that most of these checks and tests both validate and verify parts of the datawarehouse. The tests and checks have been divided into three types:

4. *Syntax*, this checks if the datawarehouse has been constructed according to the rules.
5. *Semantics*, this checks if the datawarehouse is meaningful and can be interpreted in its context.
6. *Pragmatics,* this checks if the datawarehouse represents the relevant elements of reality. This shows if it is useful given the goals and requirements.

Syntax

The syntax is checked by the environments used to create the datawarehouse components: SSIS, SQL Server and the Java Programming language. These make sure the no programming errors, typing errors or structural errors are made – these environment produce errors when such errors are made, be them meaningful or not. Explicit syntax checking was using in the ETL processes. In these cases the state of objects is validated and the prepositions in processes in order to prevent illegal states. These checks were especially useful during the development process. These checks ensured that the datawarehouse components were syntactically correct.

Semantics

The structure of the data warehouse must match with the conceptual models of the system, for example the barge dimension matches the barge object and the call activity matches the activity fact. A structural comparison has been made between both models and they match.

Because of INITI8's extensive experience in logistics their expertise has been used to find any gaps in the structure of the datawarehouse. The most important result was that more focus has been put into the measurement of distances covered and the performances delivered. This shift in focus was approved by BCTN in personal communications.

Figure 59 CarrierWeb GPS Error

The results of the input processes, which are filled data warehouse tables, have been crosschecked with reports coming from modality (CTT 2008) and with data gathered by INITI8 during their VITO MIS project. The match between the reports from modality and the transport and activity fact showed a good match. It also showed that the data for WIT is less reliable before 2006, this was confirmed by BCTN, and this restricts analysis over time between now and 2006. The comparison between the loaded data and the data gathered by INITI8 showed some minor deviations, but these can be explained by the lack of information during the planning process and strategic behavior. This verifies and validates that the correct data has been retrieved from the data source and verifies that the input process yields the correct results.

By analyzing the (intermediate) results of ETL processes a few modifications to the ETL have been made. Container transports are not always registered correctly, for example the inland terminal used of the transport is sometime omitted, and measures have been taken to fix this in the *Create Transports* and the *Create Barge Activities* steps of the ETL. The quality of coordinates produced by the CarrierWeb devices can vary, the coordinates can make large (infeasible) jumps and sometimes the devices stop working. See for example Figure 59 which depicts the path taken by a barge visiting Rotterdam (in yellow), notice the jump made (marked by the white circle). This has been corrected by adding a post processing step and extra logic to the *Create Barge Visits* Transformation. The checks ensured that the results produced by the datawarehouse are semantically correct.

Pragmatics

The datawarehouse has been extensively used as a data source during the this research. This showed that the datawarehouse produces valid results for such an application. The datawarehouse hasn't been used by BCTN yet, this is a future step which can increase the usability of (and the confidence in) the datawarehouse.

To test the pragmatics of the datawarehouse even more the requirements set have been evaluated. Table 15 shows the result of this evaluation, it describes how the datawarehouse components help in achieving the requirement and give a score ('-' = bad, 0 = 'neutral' & '+' = good) to indicate which requirements are met and which need work.

Table 15 Evaluation Requirements

Id	Name	Evaluation	Score
1	Usability	The datawarehouse has been explicitly designed around these requirements. It is able to store all relevant data.	+
2	Accuracy	The data in the datawarehouse is accurate and remains accurate as long as the assumptions made in the ETL processes remain valid. This needs be monitored.	0
3	Traceability	The ETL stores metadata for all traceable tables. Only the barge activities cannot really be tracked because they are can be executed for multiple operators at the same time.	+
4	ACID	The database environment, Microsoft SQL Server 2005, supports this by nature.	+
5	Accessibility	By using both a database backend and a data cube the data can be easily accessed using front-end tools.	+
6	Integration	The ETL processes and other components can be used without any modification for the use with other instances of Modality. Some work will be needed to adapt the ETL process to different software – this depends on the structure of the software.	+
7	Responsiveness	The database engine used has ample capacity to deal with the queries it needs to answer.	+
8	Scalability	The database environment, Microsoft SQL Server 2005, enables extreme scalability by design: it can use multiple processors and can be used in clusters. The ETL processes are made to work with every instance of Modality. There should be no problem in expanding the number of operators using the datawarehouse from a technical point of view.	+

TABLES BARGE OPERATOR DATAWAREHOUSE
This section describes the tables in the barge operator datawarehouse.

Table	Description
dim_activity_type	The activity type dimension describes the different types of activity possible.
dim_address	The address dimension describes the different actors who can be involved in the transportation process.
dim_barge	The dimension described the barges used for transportation of containers.
dim_booking_type	This dimension describes the type of booking which was applied to the transport.
dim_container	Container dimension describes the containers transported.
dim_container_type	Container type dimension descibes the different types of containers transported.
dim_region	Region describes the geographical component of transport and activities.
dim_region_filter	Region filter dimension used to filter for a given set of regions.
dim_region_filter_link	Many-to-Many filter used for the regions, this allows us to create overlapping sets of regions to filter.
dim_region_type	Region type dimension describes the type of a region.
dim_source	Source dimension, this dimension determines the provenance of data in the datawarehouse.
dim_time	Time dimension describes time related data, has a granuality of a day.
fact_activity	Barge activity fact. This fact table describes the most common barge activities
fact_transport	Container transport. This fact table describes the container transported by the barge operators

BARGE OPERATOR DATAWAREHOUSE FIELDS & SOURCES
This section describes the fields in the barge operator datawarehouse. The sources of the fields are also described.

Table	Column	Datatype	Description	Source	Source Table	Source Column
dim_activity_type	activity_type_id	Int	Primary key of the activity type.	ETL	GENERATED DURING ETL	
dim_activity_type	activity_type_name	NVarChar (50)	Name of the activity type (i.e. call or hop)	ETL	GENERATED DURING ETL	
dim_region	region_id	Int	Primary key of the region.	MAP	GENERATED ID	
dim_region	region_code	NVarChar	Code of the region	MAP	NAME POSTFIX	

		(50)			
dim_region	region_name	NVarChar (50)	Name of the region	MAP	MANUALLY ASSIGNED
dim_region	region_description	NVarChar (256)	Description of the region.	MAP	NAME POSTFIX
dim_region	region_type_id	Int	Type of the region (Links to dim_region_type)	MAP	NAME PREFIX
dim_region	parent_region_id	Int	Parent region of the region (Self reference, for example Terminal -> Port)	MAP	BASED ON CONTAINING POLYGON
dim_region_filter	region_filter_id	Int	Primary key of the region filter.	GENERATED	GENERATED USING QUERY
dim_region_filter	region_filter_name	NVarChar (50)	Name of the region filter.	GENERATED	GENERATED USING QUERY
dim_region_filter	region_filter_description	NVarChar (256)	Description of the region filter.	GENERATED	GENERATED USING QUERY
dim_region_filter_link	region_filter_id	Int	Link to dim_region_filter.	GENERATED	GENERATED USING QUERY
dim_region_filter_link	region_id	Int	Link to dim_region.	GENERATED	GENERATED USING QUERY
dim_time	time_id	Int	Primary key of the time dimension	GENERATED	GENERATED USING QUERY
dim_time	time_instance	DateTime	Instance of the current time.	GENERATED	GENERATED USING QUERY
dim_time	year_id	Int	Unique id of the current time's year.	GENERATED	GENERATED USING QUERY
dim_time	year_name	NChar (5)	Name of the current time's year.	GENERATED	GENERATED USING QUERY
dim_time	quarter_id	Int	Unique id of the current time's quarter.	GENERATED	GENERATED USING QUERY
dim_time	quarter_name	NChar (5)	Name of the current time's quater.	GENERATED	GENERATED USING QUERY
dim_time	month_id	Int	Unique id of the current time's month.	GENERATED	GENERATED USING QUERY
dim_time	month_name	NChar (10)	Name of the current time's month.	GENERATED	GENERATED USING QUERY
dim_time	week_id	Int	Unique id of the current time's week.	GENERATED	GENERATED USING QUERY
dim_time	week_name	NChar (5)	Name of the current time's week.	GENERATED	GENERATED USING QUERY

dim_time	day_id	Int	Unique id of the current time's day.	GENERATED	GENERATED USING QUERY	
dim_time	day_name	NChar (10)	Name of the current time's day.	GENERATED	GENERATED USING QUERY	
fact_activity	activity_id	Int	The primary key of the barge activity	ETL	GENERATED ID	
fact_activity	activity_type_id	Int	The type of the barge activity (hop or call - links to dim_activity_type)	ETL	ASSIGNED DURING ETL	
fact_activity	barge_id	Int	The barge executing the activity (Links to barge_id)	MODALITY	VOYAGE	BARGE
fact_activity	origin_terminal_region_id	Int	Terminal at which the activity starts (Links to region_id).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_FROM/VOYAGE.PORT_TO/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA
fact_activity	destination_terminal_region_id	Int	Terminal at which the activity ends (Links to region_id).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_FROM/VOYAGE.PORT_TO/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA
fact_activity	activity_distance	Float	Distance covered during the activities (0 the activity is a call)	MAP	BASED ON SHORTEST PATH	
fact_activity	activity_start_time	DateTime	Time at which the activity started.	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYAGE.DEPDATE/VOYTERM.EDATE
fact_activity	activity_start_time_id	Int	Time Key at which the activity started (Links to dime_time).	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYAGE.DEPDATE/VOYTERM.EDATE
fact_activity	activity_end_time	DateTime	Time at which the activity ended.	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYAGE.DEPDATE/VOYTERM.EDATE
fact_activity	activity_end_time_id	Int	Time Key at which the activity ended (Links to dime_time).	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYAGE.DEPDATE/VOYTERM.EDATE
fact_activity	activity_duration	Int	Duration of the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_containers	Int	Containers present on the barge AFTER the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_size	Float	Sum of the size of the containers present on the barge AFTER the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_nett_size_distance	Float	Performance required by the transports loaded on the barge AFTER the activity.	MAP	BASED ON SHORTEST PATH	
fact_activity	activity_gross_size_dista	Float	Performance delivered in	MAP	BASED ON SHORTEST	

	nce		order to transport the transports AFTER the activity		PATH & INTERMEDIATE ACTIVITIES	
fact_activity	activity_covered_size_distance	Float	Performance delivered by this activity.	MAP	BASED ON SHORTEST PATH	
fact_activity	activity_weight	Int	Weight of the container loaded on the barge after the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_loaded_containers	Int	Number of containers loaded during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_loaded_size	Float	Total size of the containers loaded during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_loaded_nett_size_distance	Float	Performance required by the transports loaded during activity.	MAP	BASED ON SHORTEST PATH	
fact_activity	activity_loaded_gross_size_distance	Float	Performance delivered in order to transport the transports loaded during the activity	MAP	BASED ON SHORTEST PATH & INTERMEDIATE ACTIVITIES	
fact_activity	activity_loaded_weight	Int	Weight of the containers loaded on the barge during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_discharged_containers	Int	Number of containers discharged during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_discharged_size	Float	Total size of the containers discharged during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_discharged_nett_size_distance	Float	Performance required by the transports discharged during activity.	MAP	BASED ON SHORTEST PATH	
fact_activity	activity_discharged_gross_size_distance	Float	Performance delivered in order to transport the transports discharged during the activity	MAP	BASED ON SHORTEST PATH & INTERMEDIATE ACTIVITIES	
fact_activity	activity_discharged_weight	Int	Weight of the containers discharged on the barge during the activity.	ETL	CALCULATED FIELD	
fact_activity	activity_capacity_size	Float	Maximum total of container sizes allowed on the barge during the activity.	MODALITY	BARGE	MAXTEU/MAX20FT

fact_activity	activity_capacity_size_duration	Float	Combination of the duration and the maximum size loaded on a barge.	ETL	CALCULATED FIELD	
fact_activity	activity_capacity_weight	Int	Maximum weight allowed on the barge during the activity.	MODALITY	BARGE	MAXWEIGHT
fact_activity	activity_costs_barge_rent	Float	Costs for renting the barge during the acitvity.	BCTN	DOCUMENTS	
fact_activity	activity_costs_barge_fuel_consumption	Float	Costs for fuel of the activity.	BCTN	DOCUMENTS	
fact_activity	activity_costs	Float	Costs of the activity.	BCTN	DOCUMENTS	
fact_transport	transport_id	Int	Primary key of the transports.	ETL	GENERATED ID	
fact_transport	transport_tare_weight	Int	Empty weight of the container.	MODALITY	UNITTYPE	TARE
fact_transport	transport_nett_weight	Int	Weight of the container's cargo	MODALITY	UNIT	NETT
fact_transport	transport_gross_weight	Int	Total weight of the container and cargo hauled during the transport.	ETL	CALCULATED FIELD	
fact_transport	container_size	Float	Size of the container.	MODALITY	UNITTYPE	TEUS
fact_transport	container_id	Int	Container used for transportation (Links to dim_container).	MODALITY	UNIT	CNTR
fact_transport	barge_id	Int	Barge that transported the transport (Links to dim_barge).	MODALITY	VOYAGE	BARGE
fact_transport	origin_terminal_region_id	Int	Origin terminal of the transport (Links to dim_region).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_FROM/VOYAGE.PORT_TO/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA
fact_transport	origin_exec_terminal_region_id	Int	Origin terminal of the transport at which the transport was actually loaded (Links to dim_region).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_FROM/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA
fact_transport	load_call_activity_id	Int	Load Call (Links to fact_activity).	MODALITY	VOYAGE	VOYAGE
fact_transport	destination_terminal_region_id	Int	Destination terminal of the transport (Links to dim_region).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_TO/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA

fact_transport	destination_exec_terminal_region_id	Int	Destination terminal of the transport at which the transport was actuall discharged (Links to dim_region).	MODALITY	VOYAGE/VOYTERM	VOYAGE.PORT_TO/VOYTERM.ADDRESS/VOYTERM.ADDRESS_VIA
fact_transport	discharge_call_activity_id	Int	Discharge Call (Links to fact_activity).	MODALITY	VOYAGE	VOYAGE
fact_transport	transport_nett_distance	Float	Minimum required distance by the transport (i.e. the shortest path between origin and destination)	MAP	BASED ON SHORTEST PATH	
fact_transport	transport_gross_distance	Float	Distance covered during the transportation activities.	MAP	BASED ON SHORTEST PATH & INTERMEDIATE ACTIVITIES	
fact_transport	transport_nett_size_distance	Float	Performance required by the transport.	MAP	BASED ON SHORTEST PATH	
fact_transport	transport_gross_size_distance	Float	Performance delivered in order to complete the transport.	MAP	BASED ON SHORTEST PATH & INTERMEDIATE ACTIVITIES	
fact_transport	transport_begin_time_id	Int	Key of the time at which the transport commenced. (Links to dim_time).	MODALITY	VOYAGE/VOYTERM	VOYAGE.DEPDATE/VOYTERM.EDATE
fact_transport	transport_begin_time	DateTime	Time at which the transport commenced.	MODALITY	VOYAGE/VOYTERM	VOYAGE.DEPDATE/VOYTERM.EDATE
fact_transport	transport_end_time_id	Int	Key of the time at which the transport ended (Links to dim_time).	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYTERM.EDATE
fact_transport	transport_end_time	DateTime	Time at which the transport ended.	MODALITY	VOYAGE/VOYTERM	VOYAGE.ARRDATE/VOYTERM.EDATE
fact_transport	transport_duration	Int	Duration of the transport.	ETL	CALCULATED FIELD	
fact_transport	transport_costs	Float	Revenue that the transport generated.	BCTN	DOCUMENTS	
fact_transport	booking_client_id	Int	Client for which the transport was booked (Links to dim_address).	MODALITY	BOOKING	ADDRESS_CLIENT
fact_transport	booking_shipcom_id	Int	Shippingcompany that booked the transport (Links to dim_address).	MODALITY	BOOKING	ADDRESS_SHIPCOM
fact_transport	booking_agent_id	Int	Agent that booked the transport (Links to	MODALITY	BOOKING	ADDRESS_AGENT

			dim_address).			
fact_transport	booking_type_id	Int	Type of the booking the transport is a part of (Links to dim_booking_type).	MODALITY	BOOKING	BOOKTYPE
fact_transport	booking_time_id	Int	Time key at which the transport was booked (Links to dim_time).	MODALITY	BOOKING	BOOKDATE
fact_transport	booking_time	DateTime	Time at which the transport was booked.	MODALITY	BOOKING	BOOKDATE
fact_transport	source_id	Int	Data source of the transport.	ETL	ASSIGNED DURING ETL	
dim_address	address_id	Int	Primary key of the address.	ETL	GENERATED ID	
dim_address	source_id	Int	Data source of the address data (Note that multiple instance exists and that they are merged during ETL).	ETL	ASSIGNED DURING ETL	
dim_address	address_code	NChar (20)	Code of the address.	MODALITY	ADRESS	CODE
dim_address	address_description	NVarChar (100)	Description of the address.	MODALITY	ADRESS	NAME
dim_address	address_extra	NVarChar (100)	Extra information on the address.	MODALITY	ADRESS	NAME2
dim_barge	barge_id	Int	Primary key of the barge.	ETL	GENERATED ID	
dim_barge	source_id	Int	Data source of the barge data (Note that multiple instance exists and that they are merged during ETL).	ETL	ASSIGNED DURING ETL	
dim_barge	barge_code	NChar (10)	Code of the barge.	MODALITY	BARGE	CODE
dim_barge	barge_europa_nr	Int	Europa number of the barge.	MODALITY	BARGE	EUROPANR
dim_barge	barge_name	NVarChar (50)	Name of the barge.	MODALITY	BARGE	DESCR
dim_barge	barge_max_weight	Int	Maximum total weight of containers loaded on the barge.	MODALITY	BARGE	MAXWEIGHT
dim_barge	barge_max_teu	Int	Maximum number of standard containers allowed on the barge.	MODALITY	BARGE	MAXTEU/MAX20FT
dim_barge	barge_max_feu	Int	Maximum number 40 feet	MODALITY	BARGE	MAX40FT

			containers allowed on the barge.			
dim_barge	barge_length	Float	Length of the barge.	MODALITY	BARGE	LEN
dim_barge	barge_width	Float	Width of the barge.	MODALITY	BARGE	WIDTH
dim_barge	barge_draft	Float	Maximum draft of the barge.	MODALITY	BARGE	DRAFT
dim_barge	barge_is_combination	Bit	Flag to indicate that the barge is actually a combination (i.e. multiple hulls)	MODALITY	BARGE	COMBI
dim_booking_type	booking_type_id	Int	Primary key of the booking type of the container.	ETL	GENERATED ID	
dim_booking_type	booking_type_code	NChar (2)	Code of the booking type.	MODALITY	BOOKTYPE	CODE
dim_booking_type	booking_type_name	NVarChar (50)	Name of the booking type.	MODALITY	BOOKTYPE	DESCR
dim_container	container_id	Int	Primary key of the container.	ETL	GENERATED ID	
dim_container	container_code	NVarChar (20)	ISO code of the container.	MODALITY	UNIT	CNTR
dim_container	container_type_id	Int	Type of the container type (links to the dim_container_type).	MODALITY	UNIT	UNITTYPE
dim_container_type	container_type_id	Int	Primary of th container type.	ETL	GENERATED ID	
dim_container_type	container_type_code	NVarChar (4)	Code used for the container type.	MODALITY	UNITTYPE	CODE
dim_container_type	container_type_description	NVarChar (100)	Meaningfull description of the container type.	MODALITY	UNITTYPE	DESCR
dim_container_type	container_type_tare	Int	Empty weight of the container type.	MODALITY	UNITTYPE	TARE
dim_container_type	container_type_height	Float	Height of the container type.	MODALITY	UNITTYPE	HEIGHT
dim_container_type	container_type_width	Float	Width of the container type.	MODALITY	UNITTYPE	WIDTH
dim_container_type	container_type_length	Float	Length of the container type.	MODALITY	UNITTYPE	LENGTH
dim_container_type	container_type_teus	Float	Size of the container type in TEUs.	MODALITY	UNITTYPE	TEUS
dim_container_type	container_type_reefer	Bit	Flag to indicate that the container type is a reefer.	MODALITY	UNITTYPE	REEFER
dim_container_type	container_type_iso_code	NVarChar	ISO code of the container	MODALITY	UNITTYPE	CNTRISO

		(4)	type.		
dim_source	source_id	Int	Primary key of the source table.	ETL	GENERATED ID
dim_source	source_name	NChar (10)	Name of the data source.	ETL	GENERATED DURING ETL
dim_source	source_insert	DateTime	Date at which the first insert for the source took place.	ETL	GENERATED DURING ETL
dim_source	source_update	DateTime	Date at which the last update for the source took place.	ETL	GENERATED DURING ETL
dim_region_type	region_type_id	Int	Primary key of the region type table.	GENERATED	GENERATED USING QUERY
dim_region_type	region_type_code	NVarChar(50)	Code of the region type.	GENERATED	GENERATED USING QUERY
dim_region_type	region_type_name	NVarChar(50)	Name of the region type.	GENERATED	GENERATED USING QUERY

APPENDIX E.DATA GENERATION METHODOLOGY

This appendix describes how the experiments used for evaluating bundling have been generated.

INPUT

The generation method uses the following input to create an experiment:

1. A set of barges. These barges must have an initial position at which they start. The barges availability is defined by a time window. Plans for a barge's route must start at the initial position and must remain within the time window given.
2. A flow (a pair of an origin and a destination terminal) distribution. This distribution describes the flows the experiment should use. Roulette wheel selection is used to select the flows, the larger the volume of the flow the larger the chance is that the flow gets selected. A flow distribution is easily obtained from barge operator data (using the datawarehouse) by using the count of transports for each flow as the chance it can be selected.
3. A container type distribution. The container type distribution describes the container types the experiment should use. The selection of the a container type is similar to the selection of a flow.

EXPERIMENT GENERATION

The main variable used for experiment generation is the number of containers to transport, k. An experiment consists of a number of empty routes (one for each barge) and a number of container transports. The following steps are taken during the generation process:

1. Create an empty route for each barge starting with a call at the initial position at the start of its availability.
2. Draw a container flow from the flow distribution.
3. Draw a container type from the container type distribution.
4. Create a transport based on the flow and type.
5. Repeat steps 2-4 until k containers have been generated.

The result of this is an almost unplanned experiment.

INITIAL SOLUTION

For single operator planning plans needed to be made based on the experiments. A constrained version of the ALNS algorithm was used to create these experiment plans. The main constraint is rather obvious: the planning algorithm is only allowed to transport containers for an operator using its own capacity.