An approach for designing
robotized marine container terminals

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 21 december 2004 om 10.30 uur

door Yvo André SAANEN
bestuurskundig ingenieur
geboren te ’s Gravenhage.
Dit proefschrift is goedgekeurd door de promotoren:

Prof. ir. J.C. Rijsenbrij
Prof. dr. H.G. Sol

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter

Prof. ir. J.C. Rijsenbrij, Technische Universiteit Delft, promotor
Prof. dr. H.G. Sol, Technische Universiteit Delft, promotor
Prof. dr. ir. R. Dekker, Erasmus Universiteit Rotterdam
Dr. dipl. Ing. M. Dobner, Gottwald Port Technology, Germany
Prof. dr. A. Van Harten, Universiteit Twente
Prof. dr. P. Keen, Technische Universiteit Delft
Prof. dr. ir. A. Verbraeck, University of Maryland, United States of America

© Y.A. Saanen, all right reserved.

ISBN: 90-5638-129-6
# Table of contents

**Foreword**

**Summary**

1 Container sector on the move .................................................................1
   1.1 The container terminal as indispensable link in the global supply chain......... 1
   1.2 The container terminal as a material handling system............................... 4
   1.3 Trends affecting marine container terminal business .................................. 6
   1.4 Trends (1): Increase in scale of maritime shipping ................................... 8
   1.5 Trends (2): Robotization and automation .................................................. 16
   1.6 Trends (3): Towards more dedicated terminals ......................................... 18
   1.7 Design challenges with regard to robotized terminals .............................. 21
   1.8 Outline of the thesis.................................................................................. 25

2 Review of container terminal design practices ..................................... 27
   2.1 Case experience and literature review.......................................................... 27
   2.2 Functional description of a container terminal .......................................... 27
       2.2.1 Functions of a container terminal ....................................................... 27
       2.2.2 Processes at a container terminal ....................................................... 29
   2.3 Hardware at a container terminal ............................................................... 31
       2.3.1 Introduction .......................................................................................... 31
       2.3.2 Transhipment from vessel to quay and vice versa ............................... 31
       2.3.3 Internal transportation ......................................................................... 33
       2.3.4 Yard handling ...................................................................................... 33
   2.4 Software at a container terminal ............................................................... 34
   2.5 Classification and assessment of container terminals ............................. 34
       2.5.1 Design indicators to classify various terminal types ......................... 36
       2.5.2 Performance indicators from a customer point of view .................... 40
       2.5.3 Performance indicators of terminal utilisation and productivity ........ 42
       2.5.4 Performance indicators of productivity of container handling equipment 44
       2.5.5 Performance indicators for land use efficiency ................................. 47
An Approach for Designing Robotized Marine container terminals

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.6 Container terminal operating costs</td>
<td>48</td>
</tr>
<tr>
<td>2.6 Design of a container terminal</td>
<td>50</td>
</tr>
<tr>
<td>2.6.1 Relevant trends</td>
<td>50</td>
</tr>
<tr>
<td>2.6.2 Three types of design processes</td>
<td>52</td>
</tr>
<tr>
<td>2.6.3 Design process steps</td>
<td>55</td>
</tr>
<tr>
<td>2.6.4 Aspects of design-engineering approaches</td>
<td>57</td>
</tr>
<tr>
<td>2.7 Lessons learnt: The design process of the ECT-DSL terminal</td>
<td>58</td>
</tr>
<tr>
<td>2.7.1 Project organisation and activities</td>
<td>59</td>
</tr>
<tr>
<td>2.7.2 Use of simulation models</td>
<td>60</td>
</tr>
<tr>
<td>2.7.3 Experiences from the design process</td>
<td>61</td>
</tr>
<tr>
<td>2.8 Observations from recent cases</td>
<td>63</td>
</tr>
<tr>
<td>2.8.1 Gap between functional design and consecutive design steps</td>
<td>63</td>
</tr>
<tr>
<td>2.8.2 Non-integrated design of hardware and software</td>
<td>63</td>
</tr>
<tr>
<td>2.8.3 Insufficient attention for interaction between operator and automated system</td>
<td>63</td>
</tr>
<tr>
<td>2.8.4 Gap between strategic and operational performance indicators</td>
<td>64</td>
</tr>
<tr>
<td>2.8.5 Lack of tools to analyse operations at robotized container terminals</td>
<td>64</td>
</tr>
<tr>
<td>2.8.6 Unavailability of COTS solutions for robotized terminals</td>
<td>64</td>
</tr>
<tr>
<td>2.8.7 Lacking integration between performance and cost analysis</td>
<td>64</td>
</tr>
<tr>
<td>2.8.8 Design-engineering process ends after go-live</td>
<td>66</td>
</tr>
<tr>
<td>2.9 Research approach, boundaries, and instruments</td>
<td>66</td>
</tr>
<tr>
<td>2.9.1 A simulation approach as starting point</td>
<td>66</td>
</tr>
<tr>
<td>2.9.2 Research boundaries</td>
<td>69</td>
</tr>
<tr>
<td>2.9.3 Research instruments</td>
<td>71</td>
</tr>
<tr>
<td>3 Essentials of an approach for designing robotized container terminals</td>
<td>75</td>
</tr>
<tr>
<td>3.1 Characteristics of a container terminal</td>
<td>75</td>
</tr>
<tr>
<td>3.1.1 Conflicting requirements</td>
<td>75</td>
</tr>
<tr>
<td>3.1.2 Influence of external events and circumstances</td>
<td>76</td>
</tr>
<tr>
<td>3.1.3 Information about external processes is poor</td>
<td>76</td>
</tr>
<tr>
<td>3.1.4 Multiple interaction points between system and external world</td>
<td>77</td>
</tr>
<tr>
<td>3.1.5 Processes with large variations in duration</td>
<td>77</td>
</tr>
<tr>
<td>3.1.6 Interlinked processes</td>
<td>78</td>
</tr>
<tr>
<td>3.1.7 Non-linear interactions between processes</td>
<td>79</td>
</tr>
<tr>
<td>3.1.8 Planning and co-ordination of terminal processes</td>
<td>80</td>
</tr>
<tr>
<td>3.2 Characteristics of a container terminal design process</td>
<td>81</td>
</tr>
<tr>
<td>3.3 Framework for a new approach for designing marine container terminals</td>
<td>83</td>
</tr>
</tbody>
</table>

Table of contents
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1</td>
<td>Main design activities and their basic properties</td>
<td>83</td>
</tr>
<tr>
<td>3.4</td>
<td>Starting points of the design philosophy</td>
<td>87</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Starting points of the design philosophy</td>
<td>87</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Use an object-oriented world-view</td>
<td>89</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Apply a holistic, layered view on the terminal processes</td>
<td>89</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Mirror the real system’s architecture into the model’s architecture</td>
<td>89</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Take uncertainty and process variability into account</td>
<td>90</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Use the operational processes as leitmotiv for the design</td>
<td>90</td>
</tr>
<tr>
<td>3.4.7</td>
<td>Integrate the design of manual operations and automated operations</td>
<td>91</td>
</tr>
<tr>
<td>3.4.8</td>
<td>Integrate hardware and software design</td>
<td>92</td>
</tr>
<tr>
<td>3.4.9</td>
<td>Make comprehensive and measurable objectives to assess the design</td>
<td>92</td>
</tr>
<tr>
<td>3.4.10</td>
<td>Base the decisions within the design process on performance measurements</td>
<td>93</td>
</tr>
<tr>
<td>3.4.11</td>
<td>Continue monitoring and measuring after commissioning</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>Way of working when designing robotized container terminals</td>
<td>95</td>
</tr>
<tr>
<td>4.1</td>
<td>A simulation approach throughout the design process</td>
<td>95</td>
</tr>
<tr>
<td>4.2</td>
<td>Functional design of a robotized container terminal</td>
<td>96</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Deliverables of functional design activities</td>
<td>96</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Terminal functionality, container throughput, and terminal performance</td>
<td>97</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Water- and landside handling capacity, and storage capacity</td>
<td>99</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Model system of major terminal design parameters</td>
<td>104</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Terminal geometry</td>
<td>108</td>
</tr>
<tr>
<td>4.2.6</td>
<td>Design of logistical control concept</td>
<td>111</td>
</tr>
<tr>
<td>4.2.7</td>
<td>Design of handling system</td>
<td>116</td>
</tr>
<tr>
<td>4.2.8</td>
<td>Model system of the handling system, logistical concept and layout</td>
<td>124</td>
</tr>
<tr>
<td>4.2.9</td>
<td>Design of equipment</td>
<td>128</td>
</tr>
<tr>
<td>4.2.10</td>
<td>Result after functional design</td>
<td>128</td>
</tr>
<tr>
<td>4.3</td>
<td>Technical design of a robotized container terminal</td>
<td>129</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Design issues, objectives and decisions</td>
<td>129</td>
</tr>
<tr>
<td>4.3.2</td>
<td>A simulation approach during the technical design</td>
<td>130</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Results after technical design</td>
<td>132</td>
</tr>
<tr>
<td>4.4</td>
<td>Implementation and realisation</td>
<td>133</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Design issues, objectives and decisions</td>
<td>133</td>
</tr>
<tr>
<td>4.4.2</td>
<td>A simulation approach during implementation and realisation</td>
<td>135</td>
</tr>
<tr>
<td>4.5</td>
<td>Commissioning and Operations</td>
<td>138</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Design issues, objectives and decisions</td>
<td>138</td>
</tr>
<tr>
<td>4.5.2</td>
<td>A simulation approach during commissioning and operation</td>
<td>139</td>
</tr>
</tbody>
</table>
# Table of contents

5  **Way of modelling when designing robotized container terminals**...... 141

5.1  **Overview of simulation models**................................................................. 141

5.2  **Architecture of the model suite for container terminal design**........... 141
  
  5.2.1  Functional structure 142
  
  5.2.2  Polymorphism 144
  
  5.2.3  Three layers of control 144
  
  5.2.4  Asynchronous communication between components 145

5.3  **Model suite for robotized container terminal design**.......................... 146
  
  5.3.1  Overview of models 146
  
  5.3.2  Berth Process Simulation 147
  
  5.3.3  Handling system simulation 149
  
  5.3.4  Equipment simulation 156

5.4  **Model coherence**................................................................................. 156

5.5  **Visualisation**....................................................................................... 158

5.6  **Verification and validation**................................................................. 159

6  **Way of controlling when designing robotized container terminals**..... 161

6.1  **A concurrent simulation group**................................................................. 161

6.2  **Integrating the simulation approach into the design process**............ 162
  
  6.2.1  Modelling as indispensable activity within the design process 162
  
  6.2.2  Actor interaction in system development 163
  
  6.2.3  The designer 169

6.3  **Concluding remarks**............................................................................. 170

7  **Application: the redesign of a process control system**...................... 171

7.1  **Application of design approach in real-life case**............................... 171

7.2  **Scope and objectives of the PCS replacement project**...................... 171

7.3  **Design approach entire project**.............................................................. 173
  
  7.3.1  Process activity plan 173
  
  7.3.2  Application of a simulation approach 174
  
  7.3.3  Role of the researcher in the design process 174

7.4  **Analysis of the “old PCS”**.................................................................. 174
  
  7.4.1  Approach 174
8.6.4  Integrated hard- and software design to accomplish cost-efficient solutions 222
8.6.5  Concluding remarks 222

9  Reflection, Conclusions and outlook ........................................ 225

9.1  Experiences with a simulation approach throughout the design process ........................................ 225

9.2  Expert survey ................................................................................................................................. 226
  9.2.1  Background respondents .................................................. 226
  9.2.2  Answers to questions regarding simulation approach .......... 227
  9.2.3  Findings and conclusions expert survey ............................. 229

9.3  Research findings ............................................................................................................................ 229
  9.3.1  Findings with regard to the research questions ..................... 229

9.4  Reflection on the design approach .................................................................................................. 233
  9.4.1  Experiences with the application of a simulation approach as ‘leitmotiv’ .................................. 233
  9.4.2  Experiences with regard to the guidelines ................................................................. 235
  9.4.3  Contribution to the field of research .................................................................................... 237

9.5  Future outlook and further research .............................................................................................. 238

10  Appendices ................................................................................................................................. 241

10.1  References ............................................................................................................................... 241

10.2  Interviews Delta Sealand terminal ECT (1982-1992) ................................................................ 261
  10.2.1  Introduction ......................................................................................................................... 261
  10.2.2  ECT PCS 1 Interview with Ron de Waard on 20 December 2001 ........................................... 261
  10.2.3  ECT PCS 2 Interview with Peter Metsch on 25 January 2002 ................................................ 264
  10.2.4  ECT Results Interview with Hans Vecke on 9 February 2002 ................................................. 270
  10.2.5  ECT PCS 3 Interview with Rene Besselink on 18 February 2002 ............................................ 275
  10.2.6  ECT PCS 6 Interview with Ruud van der Ham on 21 February 2002 ...................................... 279

10.3  Questions remote survey ............................................................................................................ 285

10.4  Respondents remote survey ........................................................................................................ 290

10.5  Samenvatting (in Dutch) ............................................................................................................. 291

10.6  Curriculum Vitae Yvo A. Saanen .............................................................................................. 303
Foreword

The paradox of a foreword is that it is written as final piece; when everything is ready - at least within a state that is acceptable to everyone involved – a foreword has to be written. A foreword is the place to thank people, and I certainly would like to thank some people; people who tolerated that the time and effort put into this thesis was not available for other activities (Nienke, Liam, Klaas-Pieter) – either business or pleasure; but also people that supported and encouraged me doing the research (Alexander Verbraeck, Henk Sol, Joan Rijsenbrij). Even some sceptical sounds about when it finally would be ready, provided a certain motivation to carry on and work towards completion.

Writing a dissertation is an individual thing; it is your own research even when you are part of a bigger whole. The holistic approach, crucial in the approach I propose in this thesis, is seldom applied (and certainly not realised) in large research programmes, because you are on your own, and you try not to be distracted by others or other side tracks that may not contribute to your own research.

This research leans heavily on the projects I carried out over the last 6 years at container terminals. These projects were performed within a team, and therefore I’d like to thank (in random order) Mathias Dobner, Heinz Eichner, Arjen de Waal, Pascal Bierhuizen, Onno ter Elst, Wijnand Visser, Armin Wieschemann, Paul Risnma, Gerlinde Pohlmann, Frieda de Fockert, Ruud van der Ham, Kent Busk, and Laurids Ugvlig for the contributions to my research during those projects. I also want to thank Jan-Willem Hekman who has contributed to this thesis by working on the design approach. I sincerely hope you all can find yourselves in the approach described here.

Summary

International transportation is rapidly growing. Even during the recent recession, trade growth percentages world-wide exceeded 5-10%. As a consequence, container handling capacity is rapidly growing as well. Larger vessels, bigger container terminals: it appears to be an on-going process of which the limits have not yet been reached.

At the same time, there is an increasing emphasis on cost and environmental control, which forces terminal operators to search for innovative solutions. Solutions that require less space and cost less per handled container. Here robotization and automation come into play, as they allow reducing labour by a significant amount, and allow decreasing the space usage of a terminal by percentages up to 50%.

However, this comes at a cost: the terminals that have implemented large-scale robotization and automation, have suffered from lower productivity than aimed for, as well as significant start-up problems. Many of these problems are on behalf of the terminal control software, as case research has shown us. In detail, we analysed the establishment of the ECT-DSL terminal in Rotterdam, which among others showed that:

- The occurrence rate of system failures had been underestimated, which led to inefficient recovery procedures.
- The time pressure in the project led to a focus on getting the system to run, instead of realising the functional specifications. This caused much of the specified functionality not to be implemented.
- The interfaces between various control system components were a result of a negotiation process between various design groups, instead of a rational architecture design.
- The terminal was used in a different way than planned by the terminal operator.
- The terminal was not designed from a holistic point of view, which led to sub-optimisation and components that did not work properly together.

Besides, literature and other recent case studies taught us that:

- A large gap exists between functional design of automated terminals and the technical design and software realisation.
There is a lack of interaction between the design of robotized equipment and its control software, leading to sub-optimisation of each component. Even the equipment design is fragmented, which leads to different solutions for similar problems.

Too little attention is paid to the interaction between the operator of the automated system and the system.

A gap exists between aggregate, strategic targets, like throughput volumes and vessel service times, and operational, day-to-day, hour-to-hour operational targets, such as quay crane productivity and truck service times.

There are no tools available to provide insight into the operation of automated equipment and/or automated terminals, including solutions for process control systems.

A common-off-the-shelf process control system (PCS) for automated terminals does not exist (yet), which increases the risk of realising an automated terminal.

There is a lack of integration between cost analysis tools and performance analysis tools.

Current design approaches do not address the activities after commissioning, apart from monitoring and post-evaluation.

Thus, the container handling industry and in particular container terminals, are arrived at a point where new solutions are required; solutions that provide higher performance at lower cost and less space usage. Here, robotization and large-scale automation is the way to go. However, up to now, robotized terminal development has suffered from problems during and after the design-engineering process. Therefore, we defined the following objective for our research:

To develop an approach for designing robotized, marine container terminals, which addresses the specific characteristics of such terminals, and considers the specific properties of terminal environments.

We have explored and pursued this objective within the boundaries of marine, robotized container terminals. In our design process, we focus on the reduction of risk of development of robotized terminals, and intend to include the innovative nature within the design-engineering process.
The research methodology follows the inductive hypothetic cycle. Based on inductive case studies we have designed a framework of guidelines, and developed based on these guidelines an approach for designing robotized container terminals. The approach is accompanied by an extensive suite of (simulation) models that we developed and applied in many projects (see Dobner et al. 1999, 2000, 2001, 2002, 2003, and 2004). The model suite has a building-block based architecture (Verbraeck et al., 2002), which basically means that each component is defined by its interface, whereas the implementation internally may vary according to the scope of the analysis for which the model is used. Throughout the design process, the interfaces remain unchanged, but the behaviour may change, independent of the implementation of other components.

The approach and model suite have been applied in a number of case studies during which we have gained experience with its applicability and contribution to the design process and the suite of models. We have also questioned an expert panel on the main guidelines about their usefulness.

As a starting point for our approach we chose a simulation approach (Sol, 1982), which has proven to be a solid problem solving approach for complex problems.

The approach we have designed is founded on four main activities within the design-engineering process:

− Functional design
− Technical design
− Implementation
− Commissioning and operations.

In each activity we propose applying a simulation approach, which means that for design and engineering issues a problem solving approach is used. This approach heavily relies on the use of models, especially simulation models. For this purpose we developed a model suite that may support the entire design-engineering process until the terminal has been commissioned. Even during operations, the model suite may be used for fine-tuning, or problem solving when the operational conditions change.

The basis of the approach consists of a framework of guidelines, which are the following:

− Using an object-oriented world-view.

Summary
Summary

− Applying a holistic, layered view on the terminal processes.
− Mirroring the real system’s architecture into the model’s architecture.
− Taking uncertainty and process variability into account.
− Using the operational processes as a leitmotiv for the design.
− Integrating the design of manual operations and automated operations.
− Integrating hardware and software design.
− Defining comprehensive and measurable objectives to assess the design.
− Basing the decisions within the design process on performance measurements.
− Continuing monitoring and measuring after commissioning.

These guidelines have been elaborated into a detailed approach, including a stepwise, iterative approach to design a terminal, supported by a model suite that can be applied during the various activities, providing a way to manage the process.

The design process consists of the following main steps:

− Defining the function of the terminal, the throughput capacity, and the services the terminal should provide.
− Designing the terminal’s key components, i.e. quay wall length, terminal geometry, layout of the stack, handling system, and logistical control concept.
− Designing the equipment and the process control system.

After the functional design of the terminal's components, they can be further detailed into the technical design (and specification). Subsequently, they are built (hardware) and implemented (software).

After the implementation, commissioning takes place to verify whether every component works as it should. If this is successful, operations may start, during which a period of fine-tuning will take place.

The entire process, here described in a nutshell, is completely performed following a simulation approach. This means that in all activities the use of models to evaluate and assess the quality in terms of the objectives of the terminal – at various levels of detail.

During the functional design, the typical questions to be answered are to determine the right quay length, the required number of quay cranes, the required storage capacity, and the rail and gate capacity. Subsequently, the handling system is selected.
by assessing various alternatives, and the logistical control concept is being developed to match with the handling system.

During the technical design, the process control system (or terminal operating system – TOS) is designed at a more detailed level, prototyping control algorithms, specifying the parameters, and configuring the terminal. As for most robotized terminals, there is no common-off-the-shelf software yet; much of it has to be designed and developed.

During the realisation and implementation work, a simulation approach is used to provide a test-environment for components, i.e. equipment and software components. Because during this process, components become available piece by piece, the models may provide the remainder. This can be the rest of the TOS, or the real-life input from operators, or the equipment, or the representation of arrivals of vessels and/or trucks at the terminal. The model system provides an environment, manageable by the designers to create realistic circumstances to try out and test, especially from a performance point of view.

During commissioning and operations, a simulation approach may be used to find bottlenecks, to perform quick analyses in case something appears not to work as planned. It may also fulfil a role as yardstick for the production software, as the software should provide the same performance level as the model system under the same conditions.

Throughout this process, a suite of models is used that has an architecture similar to the real system’s architecture. This makes it possible to exchange model components with real components, and to link various modes of implementation to each other without needing to change any of them.

The approach has been applied in various cases. In the software re-design and replacement project at ECT, the approach so-far has been applied to support the functional design, technical design, and software implementation. The research can be classified as action research as we have been involved in the process ourselves. However, we have found the approach well applicable in terms of finding the solutions that contribute to the terminal’s objectives, and in terms of finding the problems in the software that may hamper performance during the operation.
Besides, the feedback from the functional design teams and the software development teams has been very positive so-far.

The second case, in which the approach has been applied, is the design of a high density stacking crane, an overhead bridge crane. Here, the simulation approach has supported an integrated design of all components of the stacking crane. The crane has been modelled – both on behalf of equipment kinematics, as well as control software rules - at a very detailed level. The result was that many components could be optimized from a holistic point of view, i.e. aiming at the crane’s performance as part of an entire terminal system. This approach has saved a significant amount of money and has led to a more productive crane.

Finally, the approach has been discussed, via a remote expert survey, with various experts on behalf of container terminal and/or the use of simulation. Most of the concepts that underpin the approach are considered a contribution to the quality of the design-engineering process.

As a result the research has led to the following conclusions with regard to the research questions:

- In order to create an effective simulation approach, the models need to represent the TOS in particular, because the functionality of the TOS is a critical success factor. Representing the TOS of an automated terminal means – as compared to the representation of the TOS at manned terminals – the modelling of the software that controls the automated equipment, i.e. routing, collision avoidance, deadlock avoidance, et cetera. On top of the representation of the functionality of the TOS, it is also important to represent the technical behaviour of the TOS, i.e. response times, asynchronous behaviour, and limited possibilities to optimize decisions.

- In order to reduce the risk of automation in terms of performance loss in the operation, the approach we propose focuses on the achievement of the performance goals throughout the process, i.e. including the development and realisation processes. Secondly, our approach proposes to test software in an early stage by linking it to a realistic test environment containing the specific dynamic elements that make an operation at a container terminal so complicated. As such, complex interaction can be found and dealt with as early
as possible. Thirdly, the simulation environment allows for testing the system including the interaction with an operator.

- In order to ensure that the insight provided by the simulation approach during the process is correct, the models need to be valid. In cases where a system of a novel nature is built, especially on behalf of the TOS, the simulation can be considered a prototype of the TOS. However, during the implementation, one has to be sure that the software is built after the prototype, either by working closely together with the software supplier, or by developing a detailed software specification combined with performance requirements based on the prototype and the achieved performance levels in the design phase.

- The design approach should be independent of the technology of the solutions that are designed, compared, and assessed, to assure a comparison in the same formats and driven by the same set of variables. The set of guidelines we propose appears valid for any container terminal design, as most variables are similar to any container terminal design project. On top of the model suite we developed as part of this research, it may require additional model development in case of a completely new container terminal design. Currently most of the common handling systems at container terminals have been covered, but as technology progresses, new systems may arise. However, the basic structure of the model environment is not likely to change, just as the function of a container terminal is not likely to change. Therefore, we may presume that the model environment will be able to depict future container terminal systems as well.

With regard to applying a simulation approach throughout the entire design-engineering process, the following concluding observations can be made:

- First, we experienced that simulation is still associated to a large extent with indicative sayings rather than accurate assessments, which would make them principally unsuited for precise tasks such as prototyping and/or testing of software. However, we argue this not to be true for the model systems that we developed during our research, which contain a high level of detail to represent operations at a container terminal with sufficient realism to provide the possibility to prototype software components. This is an approach that is – based literature review – not often followed (see e.g. Wysk, 2001), which also
contributes to the fact that a simulation approach is not commonly used for purposes and/or decisions to be taken further in the design process.

− Secondly, a simulation approach requires additional time, especially in the beginning of the project. This additional time is invested with the promise of a return on investment by means of better solutions that reduce the risk of the investment. However, the time needed, is not always available. Especially during the development of software, the time it takes to provide the feedback, may be longer than feasible, which leads to decisions made on perceptions rather than scientific analysis.

− Thirdly, the professional environment within container terminals can be characterised as primarily focussed on operations. Although we observe a trend of an increasing level of education within the managerial staff of container terminals, many positions are still occupied by people that come from operations. These people are less trained in using scientific approaches when solving problems, finding bottlenecks and taking decisions: they rather depend on observations in the operations. Although this leads to good results in many cases, the risk of taking the wrong solution, or a solution that is less effective than expected, is relatively large. Moreover, when it concerns solutions of a novel nature, as is the case in robotized container terminals, experience from the past is not always the best advisor.

In the two test cases – as well as many other projects we carried out over the last years -, we have gathered concrete experiences with most of the guidelines as proposed in our design approach:

− The approach proved applicable to new developments, terminal extensions, and terminal improvement programmes. The scope and the set of feasible solutions are determined by the type of project, but in terms of the methodology, we conclude that a simulation approach from start to finish appears to be viable. Even in projects where product-based TOS software is being implemented, many questions have to be answered, and many uncertainties concerning performance and technical robustness remain to be answered during the design and implementation process.

− A crucial contribution of a simulation approach in the context of an entire design-engineering project appears to be the see-throughs that have to be
made, when creating the functional design. This process of elaboration on specific solutions – these may be equipment specifications, but also software components or algorithms - provides not only feedback to the functional design; it also provides an outlook to the technical design in terms of a prototype. Especially when the prototype (within the simulation environment) is built in accordance with the architecture of the real system – both hardware and software -, it can provide useful input to the technical specification and the implementation.

− Said prototyping functionality by means of simulation models is of high added value because most terminals are similar, which allows for re-use of components in the model suite. Here, a building block based approach appears to be valuable, as it allows changing components internally without affecting its interface.

− Our initial choice to apply a building block based architecture of our model suite has proven to be very valuable to enable the support throughout the design-engineering process. It clearly reduces development time, and therefore reduces the response time whenever a question pops up, especially during the software development. It also allows for a stable architecture of the model system during the process, where in the beginning a more aggregate behaviour is modelled, which is more detailed in later stages. However, the interface of each component remains unchanged. For particular purposes, it may even be the case that some components are applied in a detailed implementation, whereas others are present in a more aggregated mode. This significantly reduces development time, and speeds up the experimentation.

− The use of a simulation approach fosters continuous attention for performance issues. As every project is under time pressure, it is common that during implementation/realisation the focus of the people involved – including the managerial level – moves from the functional requirements to the basic requirement of getting the terminal running. The attention for performance issues is then deliberately delayed to a later point in time. However, this may cause changing things at this later point in time to be impossible, because of decisions made to bring the system to work. When a model system is available allowing the people involved to assess whether these decisions would affect
performance, different solutions can be sought and decisions can be carried out knowing the consequences. Of course, it is essential to make sure that this model system is valid and trustworthy for these kinds of assessments.

− In addition to the previous point, the simulation approach can not only be used to keep the focus on performance (parallel to getting the system to work), but it also can be used to test the system under all kinds of operational circumstances. This means that the likelihood of huge performance losses in the case of break-downs, or other disrupting factors becomes less, herewith reducing the risk for the terminal operator. When the results of these tests are shared with the people that will operate the terminal after going live, the negative effects are likely to be smaller during the start-up of operations.

− A prerequisite to the success of the approach proposed in this research is the cooperation and coordination between the simulation group and the software engineering team. We have experienced that this process is not always easy because of two reasons. The main reason is the difference in objectives. The prototype in the simulation environment is built to assess the contribution to performance, rather than to develop a piece of software that can be used in an operational environment. Although the two can go together, it is not always the case, which may lead to a solution that works perfectly in the model system, but difficult to transfer to the production software. The second reason is the concurrency between the simulation approach and the actual software engineering and development. As the simulation approach continues during the software development, the detailed design runs on two tracks that may deviate at some points. A regular exchange of ideas and designs is required to keep it on the same track.

Finally, when looking into the future, we see a further extension of the simulation approach at container terminals. First, apart from a monitoring function, a simulation approach may also be used to support real-time decision-making. For instance, when something unexpected happens, the actual situation may be loaded into the model system, with which then multiple courses of actions can be analysed. The outcome – i.e. the best course of action - may then be fed back into the real system. Although this is principally a typical simulation cycle, the requirements to the model system are of such a nature that this kind of use is not yet trivial. Hence, it requires an on-line
interface between model system and the actual data sources (i.e. the database of the TOS), as well as requiring a short experiment lead time. At present, the models that could provide valuable decision support are still too slow; to get solid outcomes at least 8 to 24 hours of runtime is required. We expect that these kinds of applications will be used more in the future as technology will be more able to support them. A way to go here is also to apply distributed computing: all terminal operating systems of robotized terminals are running as distributed client-server applications; this may also be needed for the test-environment to cope with the required computation power. This is an additional reason for applying a building block based design of model systems, such as the model suite we developed as part of this research.

A second application that we foresee in the near future is one that we address as reverse simulation. With this notion we mean that the past is replayed for analysis purposes. For instance, all events are logged from an operation of two weeks. The log is fed into the simulation environment and replayed so that all anomalies can be analysed off-line. Then, based on the analysis resulting from the replay, parameters can be adjusted and the effect of such adjustments can be analysed by repeating the replay. This approach, using the same environment as for other problem solving cycles, will lead to the ability to configure a terminal operating system avoiding a trial-and-error approach, which inevitably leads to performance loss in some cases. Because the comparison is based on real data, and is always of a comparative nature – after all, first the settings that are logged are replayed, then the new settings are analysed – the outcome of such simulation experiments are valid for the outcomes in practice.
An Approach for Designing Robotized Marine container terminals
1 CONTAINER SECTOR ON THE MOVE

In this chapter we introduce the container handling sector: the main trends and its consequences for the (re)design and development of container terminals. We stress the critical situation for terminal operators that need to expand their capacity to cope with the ever-increasing growth, as they need to consider innovative operating modes to stay competitive. The chapter ends with the challenges we take up in this research.

1.1 The container terminal as indispensable link in the global supply chain

The Netherlands has a relatively good geographical location in terms of accessibility by air, sea, waterways, rail, and roads. The accessibility supported by extensive infrastructure has been a basis for intensive international trade since the Middle Ages. Right now, this infrastructure has evolved to a network acting as one of Europe’s gateways with the mainports Rotterdam and Amsterdam as physical centres. Common logistic activities are transportation, transhipment, and storage, but also activities such as sorting, repackaging, and final assembly are counted to the value adding logistic activities. Throughout the logistic channels, the objective is to provide service with a high quality, availability, predictability, reliability, and speed against costs that are as low as possible. Therefore, these properties become the key qualities of a logistics service provider.

A major trend, fostered by the need of efficiency improvement, in the transportation of goods is standardisation of which the maritime container is an excellent example. Compared to the overall growth of intercontinental transportation, the growth of containerised transportation is high. At present 60% of world general cargo trade is carried in containers. On trades between industrialised countries, this percentage approaches over 80% (World Bank, 2002). A standardised load unit contributes to the overall efficiency but also to the interoperability of different transportation systems and modes.

Marine container terminals are important links in the intercontinental container flow (see Figure 1-1). They provide high-speed transhipment from deepsea vessels to feeders, barges, rail, and trucks and vice versa. Besides, they provide a location where containers, full and empty, can be stored during a longer period. The latter function makes it possible to decouple the intercontinental and continental flows in time. Zijderveld (1995) confirms this by mentioning the linkage between the maritime
mode and different continental modes, which would be too complex when not decoupled in time and space.

![Image of intercontinental transportation chain]

*Figure 1-1: Example of intercontinental transportation chain (Dobner, 2001c)*

The material flow is the key process not only at a container terminal, but in other parts of the transportation chain as well. Of course the accompanying information flow is in the spotlights right now, but efficient material flow handling will remain the core process of enterprises involved in the transportation chain. Material handling systems are complicated and integrated combinations of material, machines, processes and people (Meer, 2000), generally addressable as *resources*. Since the cost of labour was relatively low in the early stages of industrial development, labour was used without too much consideration, and still is in non-western countries. Efficiency in area utilisation, material handling, and control systems was given little consideration (Meer, 2000). However, times have changed. The cost of labour has increased rapidly and the fast development of technology in handling equipment and control and communication systems encouraged enterprises to reconsider the current ratio between manned and automated processes.

This trend can also be observed at marine container terminals. Especially in Northern West Europe (see table 1-2), terminals increasingly consider *robotized* operations as a serious and viable alternative for manual operations. These terminals are mainly operated by robotized equipment and controlled by automated control systems.
In the design process of robotized terminals new design issues arise, dealing with the control of automated equipment, the design of terminal operating systems, and with the logistic concept of the terminal. As manually operated terminals are still labour intensive, the objective is to minimise the required labour hours, which influences the choice of the logistic concept. In an automated terminal, which is less labour intensive, new possibilities arise, such as automated preparation for upcoming operations. It is a challenge to utilise these possibilities to a full extent when developing robotized terminals.

A second element that makes the design of robotized terminals challenging, is the high degree of automation in combination with relatively large external and unpredictable influences. A major reason for the unpredictable nature terminal processes comes from the arrival pattern of the modes visiting the terminal – deep-sea, barge, truck, rail, and short-sea. Secondly, the arrival pattern itself is unreliable or even completely unpredictable, especially on the landside that is influenced by factors like delivery and pick-up time-windows and congestion. Thirdly, the information provision to the terminal is very poor in terms of quality and timeliness.

The complexity of a robotized container terminal also comes from the complexities in the design process. First, there are many people involved in this process; people with different backgrounds and disciplines, from different departments and usually even from different organisations. A container terminal design requires for instance knowledge of civil engineering, mechanical engineering, electrical engineering, mathematics, software engineering, cost analysis, logistics, and project management.

A second element of the process complexity is the duration of the design process. It usually takes at least five years from the first rough ideas until the final realisation and commissioning. During this long lasting process the team of people changes due to the change of tasks to be performed, or due to the regular turnover or due to other reasons. Furthermore, the content of the process changes in terms of the level of detail that increases and the design tasks evolving from a more functional character to a technical character. Finally, the environment, e.g. competitors, technology, financial situation of the enterprise, may change as well during the process.

The following sections describe the context of this research. The main theme is the design of robotized marine container terminals. First, a number of issues concerning material handling systems are described, as a container terminal’s main function is
material handling. Subsequently, we will go deeper into a number of issues regarding the design of a robotized marine container terminal. We end this chapter with the outline of this thesis.

1.2 The container terminal as a material handling system

Like a warehouse, a distribution centre, and a production plant, a marine container terminal, or more generally a transhipment terminal, can be described as a material handling system. Following Van der Meer (2000) we mention two definitions of material handling systems. The first one (Kulwiec, 1985) gives the following description:

“Materials handling is a system or combination of methods, facilities, labour, and equipment for moving, packaging and storing of materials to meet specific objectives.”

Tompkins et al. (1996) broaden the definition to the following:

“Material handling means providing the right amount of the right material, in the right condition, at the right place, at the right time, in the right position, in the right sequence, and for the right cost, by using the right method(s).”

Important elements within this definition from a container terminal perspective are:
- Right time since the loading and unloading operation of vessels is a time-critical operation.
- Right position, because containers have to be stored at the right position in the stack (storage area) as well as in the ship.
- Right sequence, since the container loading sequence has to be watched because of weight and safety restrictions and conformation to stowage plans.

Rijsenbrij (1999) defines a terminal as follows: “A terminal is an organisation offering a total package of activities and services to handle, store and control cargo to and from transportation modes with a balance in handling and services to the transportation modes against minimised costs.”

An organised and systematic method for the analysis of material handling systems is Systematic Handling Analysis (Muther, 1973). This method departs from three basic elements:
- The load-unit that has to be moved, for instance a box, a pallet, a container, or a piece of break-bulk.
An Approach for Designing Robotized Marine container terminals

Chapter 1: Container sector on the move

- The process sequence a load-unit has to go through, for transhipment, horizontal transportation, and storage.
- The time-horizon in which the process sequence of a certain load-unit has to take place.

![Cost vs. distance in an intercontinental transportation chain](image)

Figure 1-2: Cost distribution of an example transportation chain (see also Velsink, 1995, Dobner et al., 2001c).

Timing, positioning, and sequencing affect the costs of material handling. The notion of costs is very important here, because logistics is often not seen as a profit centre of an organisation but rather as a cost centre. Tompkins et al. (1996), estimated that material handling activities account for 20% to 50% of the total manufacturing costs as well as 15% to 70% of a product’s total cost (Meer, 2000). Costs are even more important when looking at the relative share of costs related to transhipment within the entire transportation and production process (see Figure 1-2) of which Velsink (1995) provides a generic picture. Not long ago, the share of sea shipping as part of the cost of the entire transportation was large (up to 50%). Moreover, due to an increase of scale, the relative share of the cost of sea shipping has dramatically decreased (Dobner, 2001c, Wijnolst et al., 1999). Figure 1-3 shows that in the last decade, the share of cost of sea shipping went from approximately 40% to less than 30%. Even more dramatically is the absolute decrease of the total cost of the entire chain, from approximately US$ 4,000 in 1992 to get a container from China to
Germany. Nowadays (2001) a shipper has to pay US$ 2,200, a decrease of almost 50%. However, the profits have dropped even more: from an acceptable 10% over all enterprises in the transportation chain to 2.5% today (2001). Figure 1-3 also shows the possible future. Although due to economies of scale the cost of intercontinental transportation will most likely continue to decrease, the marginal benefits will be smaller. Wijnolst (1999) estimates the cost reduction of a Malacca-max vessel (18,000 TEU) at 16% in comparison with a Port-Panamax vessel. The enterprises in the transportation chain will try to improve their efficiency, either by better adjustment to each other’s processes and/or by optimising their own processes. Therefore, we expect that in the near future the emphasis for improvement of the transportation chain will be on material handling at the terminal and inland transportation.

Figure 1-3: Changes in cost within entire transportation chain (Dobner, 2001c)

1.3 Trends affecting marine container terminal business

Global changes have a significant influence on today’s and tomorrow’s supply chains. The container terminal, as one of the main links in the intercontinental supply chain, is highly affected by the changes that we are facing. The following trends can be identified that force terminal operators on the one hand and encourage them on the other hand to reconsider their current terminal concept (Hekman, 2001; Rijsenbrij, 2001):

- The increase of volume that has to be handled within a limited period.

  Shippers and consignees require better services from the shipping lines. This
will increasingly result in Service Level Agreements between shipping lines and terminal operators. Guaranteed service times for the delivery and receipt of containers, guaranteed handling times of vessels and continental modes will be required from the terminal operator.

- The continuing demand for port facilities and the awareness of the scarcity and value of land for many destinations (industry, housing, infrastructure, recreation) will cause an increasing scarcity of land for port operations, i.e. for container terminals. This will lead to an increased need for dense operations, especially for the stacking of containers.

- Scarcity and cost of labour are increasing, making the cost balance for a (partly) automated terminal increasingly interesting. Especially the scarcity of skilled crane drivers and straddle carrier drivers, together with well-organised unions has led to excessively high wages in some places.

- The shift from a traditional working pattern (Monday – Friday, 9 am until 5 pm) to a 24-hour economy leads to a more frequent supply and a better utilisation of infrastructure. However, it also requires an improved information exchange between shifts for instance.

- The environmental regulations regarding noise and air pollution have a serious impact on the choice of handling equipment.

- Global care for health and working conditions influences terminal layout and the selection of material handling techniques.

Not only global trends affect the demands upon container terminals; technological developments, such as the increase in size of deepsea vessels, influence terminal design as well. This influence is twofold. On the one hand, these technological developments lead to an increasing demand for better performance of both terminal and terminal equipment. On the other hand, technological developments can be used to improve terminal and terminal equipment performance:

- Increasing vessel size leads to higher berth productivity and higher peak loads on the terminal (handling and storage). It also leads to a decreasing equipment utilisation, when the total volume remains equal.

- The abilities of Information and Communication Technology (ICT) allow for faster information exchange and information processing with a higher quality. This is taking place at all levels within organisations and between organisations,
and for simple and more complicated tasks. When information handling is taken over by a computer application, we address this as \textit{automation}.

- The ability to robotize equipment and to automate the equipment control. Together with the application of ICT, robotization enables terminal operators to consider other handling system concepts than traditionally feasible, both technically and economically. With \textit{robotization} we mean that physical tasks are performed by machines that are controlled by computers.

As a result of both the circumstantial trends as well as the technological trends in the area of marine container terminals, an increase of scale as well as large-scale automation is rapidly taking place (Dobner, 2000a; Rijsenbrij, 1999, 2001, 2002a). Besides huge investment costs, the redesign, extension, and new construction of container terminals require in-depth insight beforehand. The complexity of a number of interrelated (automated) systems is high and the type of mutual effects often of a non-linear nature. This makes the design process of automated container terminals an interesting theme to research. But before going into the details with regard to the design process, it is worthwhile to take a closer look at the two main drivers of automation: the increasing productivity required by the customers and the increasing labour costs, which makes automation also feasible from a financial point of view.

The increasing demands for productivity come from an increase in scale of mainly the deepsea vessels, whereas the service time-windows for container terminal visits remain equal. So in the end, these technical developments result in an increase of the required handling performance of terminals. This trend is described in more detail in section 1.4. The increase of labour cost, both absolute and relative when looking at the overall terminal cost, has initiated, and fostered the introduction of automated equipment as replacement for man-powered equipment. This trend is described in section 1.5.

1.4 Trends (1): Increase in scale of maritime shipping

The world-wide intercontinental and continental volumes of container transport are rapidly increasing as a result of globalisation, economical growth and geographical distribution of activities (Stevens, 1997). At the same time, existing and newly planned terminals are trying to attract as much volume as they can handle, which classifies the container handling sector as very competitive.
Chapter 1: Container sector on the move

It is expected (e.g. Rijsenbrij, 1999, Connekt, 2001, Drewry, 2003) that the growth will continue with an annual rate of 5 to 8% for the next couple of decades. This expectation is also underpinned by the increase of intercontinental transportation capacity, which is – according to the number of vessels in production as well as the number of ordered vessels (World Bank, 2002) – going to grow by 25% until 2003 (Connekt, 2001). This transport is carried out via different modes of transport. Large-scale deepsea vessels carry out the major part – measured in ton kilometres. Traditionally the share of maritime shipping in the total transportation cost was 50%, but due to the increase in scale, this share is decreasing to approximately 30% (Rijsenbrij, 2000).
The increasing intercontinental volume is the main reason for the increase in scale of deepsea vessels in order to accomplish higher levels of efficiency (see Figure 1-5) leading to decreasing costs per container (Wijnolst, 1999; Bruzzone et al., 1996). Nowadays, vessels carrying more than 6,500 TEU are increasingly common. For instance in the year 2000, there were more than 200 post-Panamax (4,000+ TEU) vessels in operation (Rijsenbrij, 2000) – only Maersk-Sealand already had 29 vessels in operation larger than 6,000 TEU\(^1\). Most terminals under design are laid out for vessels of 12,500 TEU. Suezmax and Malaccamax vessels – vessels that are still on the drawing boards – can even carry up to 18,000 TEU (Wijnolst, 1999; World Bank, 2002).

\[\text{Figure 1-6: Quay gantry cranes from 1960 to 2000 (Rijsenbrij, 2000)}\]

As the vessels continue to grow, the requirements to the ports and the container terminals that are handling the larger vessels continue to increase as well. For instance, the draught and the crane lifting height and outreach need to be increased (see Figure 1-6). This trend has resulted in large intercontinental hubs, called main ports. With the introduction of hub terminals, the transportation patterns change as

\(^{1}\) Data acquired at 17/03/2004 at http://www.maersk.com/.
well; the principle of hub-and-spoke is introduced: The large (mother) vessels only call on a small number of terminals (the hubs), and the final maritime distribution is done by smaller vessels, deep-sea, feeder and short-sea (the spokes). At the hubs – e.g. Singapore, Tanjung Pelepas, Gioa Tauro, Salalah, and Algeciras – mainly transhipment takes place. In addition to transhipment from mother vessels to feeder services, transhipment from mother vessel to mother vessel takes place as well.

The rationale behind the hub-and-spoke concept is the use of the right equipment for the right operation, as well as to bundle large cargo flows. Because the equipment (quay cranes) and the berth – especially the draught at the quay – required to handle the bigger vessels, are very expensive, it is not cost efficient to handle smaller vessels as well. Therefore, a system is used, which consists of very large vessels that sail between a very limited number of container terminals (the hubs), from which smaller container terminals are served by means of feeder ships (the spokes). One of the two largest container terminals in the world – Singapore – is an 80% transhipment terminal (see for definition of transhipment, Figure 2-8 on page 39), meaning that the containers that arrive at the waterside also depart at the waterside. The decision to dedicate a terminal to waterside movements influences the possibilities for terminal design and increases the potential for beefing up the waterside performance. Although the Port of Rotterdam for instance is one of the largest container terminals in the world (see Table 1-1) – in 2003 it was the largest container terminal in Europe and the eighth largest in the world – it is not a transhipment terminal: the major part (70% to 80%, source ECT) goes through the terminal in both ways. A terminal with this type of material flow is called an import-export terminal.

Table 1-1: Transhipment figures container terminals worldwide (Cargo Systems, August 2004)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hong Kong</td>
<td>1 (1)</td>
<td>East Asia</td>
<td>36%</td>
<td>20,450,000</td>
<td>18,140,000</td>
<td>17,800,000</td>
<td>17,040,000</td>
<td>15,944,793</td>
</tr>
<tr>
<td>Singapore</td>
<td>2 (2)</td>
<td>South East Asia</td>
<td>14%</td>
<td>18,100,000</td>
<td>16,800,000</td>
<td>15,520,000</td>
<td>17,040,000</td>
<td>15,944,793</td>
</tr>
<tr>
<td>Shanghai</td>
<td>3 (4)</td>
<td>East Asia</td>
<td>170%</td>
<td>11,370,000</td>
<td>8,610,000</td>
<td>6,334,000</td>
<td>5,163,000</td>
<td>4,210,000</td>
</tr>
<tr>
<td>Shenzhen</td>
<td>4 (6)</td>
<td>East Asia</td>
<td>167%</td>
<td>10,650,000</td>
<td>7,633,754</td>
<td>5,076,435</td>
<td>3,993,714</td>
<td>4,210,000</td>
</tr>
<tr>
<td>Busan</td>
<td>5 (3)</td>
<td>North East Asia</td>
<td>61%</td>
<td>10,368,000</td>
<td>9,436,307</td>
<td>7,906,807</td>
<td>7,540,387</td>
<td>6,439,589</td>
</tr>
<tr>
<td>Kaohsiung</td>
<td>6 (5)</td>
<td>East Asia</td>
<td>27%</td>
<td>8,844,000</td>
<td>8,493,000</td>
<td>7,540,000</td>
<td>7,425,832</td>
<td>6,985,361</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>7 (8)</td>
<td>US West Coast</td>
<td>87%</td>
<td>7,148,940</td>
<td>6,105,863</td>
<td>5,183,520</td>
<td>4,879,429</td>
<td>3,828,852</td>
</tr>
<tr>
<td>Rotterdam</td>
<td>8 (7)</td>
<td>Northern Europe</td>
<td>12%</td>
<td>7,118,000</td>
<td>6,500,000</td>
<td>5,844,951</td>
<td>6,300,000</td>
<td>6,343,242</td>
</tr>
<tr>
<td>Hamburg</td>
<td>9 (9)</td>
<td>Northern Europe</td>
<td>64%</td>
<td>6,138,000</td>
<td>5,373,999</td>
<td>4,689,000</td>
<td>4,250,000</td>
<td>3,738,307</td>
</tr>
</tbody>
</table>

Although the vessels are growing, the turn-around times at ports are not. The estimates (Connekt, 2000a, Ypma, 1999) give a sustained turn-around time of 24...
hours for the largest vessels. Although at first sight, one would expect an increase of the service times in ports as well but this is not likely to happen because the worldwide weekly call schemes are based on a certain time in port. Additional delays most likely mean the need for additional vessels, which makes the vessels less efficient. Therefore, the cost-benefit analyses of larger vessels are based on the assumption that the volumes carried per vessel will increase proportionally to the increase in vessel size. However, this also means that the time in port should not increase, because this would inevitably lead to the need for more vessels. The limitation of service time at the port also means that up to 9,000 TEU, i.e. the maximum expected call size for a 12,500 TEU vessel (Saanen, 2001b) has to be handled within 24 hours (see Figure 1-7).

![Diagram showing the consequences of increased container transport volumes](image)

*Figure 1-7: Consequences of increased container transport volumes (Connekt, 2000)*

With the current maximum gross waterside productivities – varying from approximately 20 to 35 gross container moves per hour per quay crane in the Hamburg-Le Havre range – it would require 10 to 15 quay cranes continuously operating on such a vessel during this 24 hour period. The use of these numbers of quay cranes at a single vessel is not feasible due to a lack of space – quay cranes need
about 30 metres distance from centre to centre. Thus, the equipment needs to be more productive instead of more numerous. The current quay cranes for instance, that are foreseen at new jumbo terminals (see e.g. FAMAS, 1999, Rijsenbrij, 2002b) are capable of performing 60 to 100 cycles per hour during a 24 hour period. The berth productivity – the production on a single vessel, measured from lines on to lines off – is expected to increase from 100 to 150 moves per hour\(^2\) (mph) – at the moment the world-wide record is 336 container moves per hour (mph) (2/2001, Kwai Chung Port Container Terminals in Hong Kong) – to future averages of 200 to 300 mph using six to eight quay cranes. That means that vessels with a call size between 4,000 and 7,000 containers can be handled within 24 hours.

This increased berth productivity requires a handling system – consisting of a horizontal transportation and a yard handling system – that can support this waterside productivity in an efficient manner. The bottleneck in increasing the quay crane performance today is the combination of horizontal transportation and yard handling (see e.g. Dobner 2001a, b, c). Whereas quay cranes are capable of more than 40 cycles per hour, the actual productivity in most terminals does not exceed 20 to 30 mph.

Most terminals with a handled volume above one million TEU use terminal tractors or straddle carriers as waterside transportation system (see Hekman, 2001). In most cases, the terminal tractor is combined with either RTGs – rubber-tyred gantry cranes – or RMGs – rail-mounted gantry cranes as a yard handling system. Up to now, the terminal tractor and RTG have not been robotized, although the AGV – automated guided vehicle – is close to an automated terminal tractor. Although the manual yard handling and horizontal transportation systems provide a lot of operational flexibility, it is questionable whether a manned way of controlling is capable of supporting a berth productivity of 200 to 300 mph. Besides, it is questionable whether the operation is still cost-efficient when the productivity is increased to these high numbers.

\(^2\) With moves we mean containers. All handling capacities are measured in moves, whereas all storage capacities are measures in TEUs, i.e. twenty-foot-equivalents. Since there can be more than one container per crane cycle (momentarily up to 4 containers, in the case of dual cycling twin-lift), it is necessary to distinguish between crane cycles, which are mostly determined by the technical crane specifications, and the moves, which mainly depend on the ability to handle multiple containers per cycle.
An increase of the berth productivity (number of moves over the quay per time-unit) leads to a lower terminal utilisation with all other parameters being equal (e.g. without a yearly volume increase), since the volume is transhipped in a smaller period. Rijsenbrij (2001) describes an example of the consequences of increasing the crane and berth productivity as shown in Table 1-2. It shows three situations of the same terminal (set-up 1, 2, and 3). As can be seen in the table, the potential yearly throughput of high productive terminals (set-up 2 and 3) are much higher than of a low productive berth. However, simply increasing the berth productivity does not lead to more volume. For instance, storage capacity limitations may block the opportunity to attract more volume. In that case of equal volumes, the quay and equipment utilisation significantly decrease to unbenevolent levels. In order to make these cranes cost-effective, the volume should at least triple.

Table 1-2: Effect of increased berth productivity on crane utilisation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type of parameter</th>
<th>Set-up 1</th>
<th>Set-up 2</th>
<th>Set-up 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Volume (TEU) per m quay length</td>
<td>Input (objective)</td>
<td>1,500</td>
<td>1,500</td>
<td>1,500</td>
</tr>
<tr>
<td>B Berth length (m)</td>
<td>End. Parameter*</td>
<td>350</td>
<td>350</td>
<td>350</td>
</tr>
<tr>
<td>C Yearly berth volume (TEU)</td>
<td>B x A</td>
<td>525,000</td>
<td>525,000</td>
<td>525,000</td>
</tr>
<tr>
<td>D Vessel size (TEU)</td>
<td>Ex. Parameter**</td>
<td>5,000</td>
<td>12,500</td>
<td>12,500</td>
</tr>
<tr>
<td>E Average call size (containers)</td>
<td>Ex. Parameter**</td>
<td>2,000</td>
<td>8,000</td>
<td>8,000</td>
</tr>
<tr>
<td>F Operational quay crane productivity (moves per hour)</td>
<td>Ex. Parameter**</td>
<td>30</td>
<td>45</td>
<td>65</td>
</tr>
<tr>
<td>G Required berth productivity (moves per hour)</td>
<td>Ex. Parameter**</td>
<td>100</td>
<td>300</td>
<td>350</td>
</tr>
<tr>
<td>H Required number of quay cranes</td>
<td>G / H</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>I Crane utilisation during vessel operation (%)</td>
<td>End. Parameter*</td>
<td>90%</td>
<td>85%</td>
<td>85%</td>
</tr>
<tr>
<td>J Gross crane hours per vessel (=net hrs. + 10%)</td>
<td>E / F</td>
<td>73</td>
<td>196</td>
<td>135</td>
</tr>
<tr>
<td>K Resulting vessel service time</td>
<td>J / (H x I)</td>
<td>20</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td>L Gross berth time (=net + 2 hrs.)</td>
<td>K + 2</td>
<td>22</td>
<td>35</td>
<td>29</td>
</tr>
<tr>
<td>M Yearly number of calls in order to handle yearly volume</td>
<td>C / E</td>
<td>263</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>N Yearly number of calls to obtain 40% berth utilisation</td>
<td>8760 x 0.4 / L</td>
<td>157</td>
<td>100</td>
<td>123</td>
</tr>
<tr>
<td>O Potential throughput over 1 berth (40% occ.) (moves pa)</td>
<td>N x E</td>
<td>313,000</td>
<td>804,000</td>
<td>982,000</td>
</tr>
<tr>
<td>P TEU factor</td>
<td>Ex. Parameter**</td>
<td>1.60</td>
<td>1.60</td>
<td>1.60</td>
</tr>
<tr>
<td>Q Potential throughput over 1 berth (40% occ.) (in TEU pa)</td>
<td>O x P</td>
<td>500,800</td>
<td>1,286,400</td>
<td>1,571,200</td>
</tr>
<tr>
<td>R Potential productivity per m/quay (based on P)</td>
<td>Q / B</td>
<td>1,431</td>
<td>3,675</td>
<td>4,489</td>
</tr>
<tr>
<td>S Realised berth utilisation with yearly volume (C)</td>
<td>I x (C / E) / 8760</td>
<td>67%</td>
<td>26%</td>
<td>21%</td>
</tr>
<tr>
<td>T Crane hours per year to handle yearly volume (C)</td>
<td>J x (C / E) / H</td>
<td>4,813</td>
<td>1,833</td>
<td>1,481</td>
</tr>
<tr>
<td>U Crane utilisation (100% = 8760 hrs/y)</td>
<td>S / 8760</td>
<td>55%</td>
<td>21%</td>
<td>17%</td>
</tr>
</tbody>
</table>

* Endogeneous parameter
** Exogeneous parameter

In Table 1-3, a benchmark of terminal performance indicators is shown according to Drewry Shipping Consultants Ltd (1998). Strangely enough, the benchmarks for quay productivity is measured in TEUs, although typically handling capacity is measured in moves (see footnote 2).
Table 1-3: Container terminal utilisation indicators (Drewry, 1998)

<table>
<thead>
<tr>
<th>Operational factor (throughput annually)</th>
<th>“At capacity” terminal benchmark</th>
<th>“Rule of thumb” industry benchmark</th>
<th>“In practice” industry sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEU throughput/metre quay</td>
<td>965</td>
<td>750</td>
<td>530</td>
</tr>
<tr>
<td>TEU throughput per yard ha</td>
<td>32,000</td>
<td>20,000</td>
<td>17,500</td>
</tr>
<tr>
<td>TEU throughput per quay crane</td>
<td>112,500</td>
<td>105,000</td>
<td>84,000</td>
</tr>
<tr>
<td>Yard gantries per quay crane</td>
<td>3.25</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Another phenomenon is the increase of scale on the landside transportation system. In Rotterdam (Delta Multi User terminal) for instance a multi-trailer system, a truck with up to five trailers, fulfills the internal transport. In Rotterdam these multi-trailers (see Figure 1-8) are also used for inter terminal transport which is mostly carried out over longer distances (10 to 15 km). Also wider crane tracks (9 to 12 containers wide), a higher stack (up to eight high in Singapore for full containers (see Figure 1-8), compared to two to four high in Rotterdam) can be classified as a trend in the development of marine container terminal handling systems. Furthermore, in the US double-stack trains are very common, as are road trains in Australia. In Europe, the transportation to and from the hinterland is rather small-scaled. Furthermore, the scale at the current terminal hardly matches with the available area: especially in the US, it is common to stack the containers one high on chassis (wheeled operation). Therefore, an increase of scale on the landside will become inevitable in the next couple of decades. One of the developments in this direction is the three TEU truck that is in a test phase in the Netherlands at the moment (2004).

Is this increase in scale likely to continue, or are there physical, economical, or technical restrictions that will soon be the limiting factor?
First, the increase in scale is not desirable from every perspective. Although the costs of maritime shipping decrease, the cost of terminal operations will show an opposite trend. When the increase in scale of vessels coincides with a proportional decrease of the call frequency (the number of times a vessel visits the terminal), and at the same time the time-windows of the operation – or vessel turn-around times – remain unchanged, the terminal efficiency decreases, because the peak productivity increases and the equipment utilisation decreases. From the point of view of the entire transportation chain, there could be a net benefit, but many terminals in the world are multi-user terminals, which means that the vessels operated are from different shipping lines and the terminal operator is yet another company. Nevertheless, even in the case that the same company – take for instance AP Møller (Maersk-Sealand as shipping line and APM Terminals as terminal operator) or P&O Nedlloyd (P&O Nedlloyd as shipping line and P&O Ports as terminal operator) – owns a shipping line and terminals, the single player’s theory is not completely applicable because within the company the profitability of different departments is measured separately.

Secondly, the increasing size of the deepsea vessels puts a high burden on the terminal's equipment (see Table 1-2). Increasing the lifting height and outreach of the ship-to-shore cranes, slows them down because of the longer distances trolley and hoist have to bridge. Furthermore, because of the increased volume at peak the landside operation has to function very well in terms of preplanning and pre-stacking. In the case that the landside supply of containers is not well controlled by the terminal operator – which is often the case – the pre-planning process is being hampered. Therefore, either complete new handling system solutions are required or the performance demands have to be adjusted to economically and technically feasible levels.

1.5 Trends (2): Robotization and automation

Due to the enormous growth of volume and due to price declines in maritime shipping (Dobner, 1999), the call for automation of container terminals, as a reliable alternative for manually-driven equipment becomes louder and louder. Recent comparisons (Dobner et al., 2000a; Connekt, 2001) clearly show a relative cost advantage in favour of robotized terminals compared to manually operated terminals. This is increasing with an increase of labour costs. In one of the cases, we found that these cost reductions were estimated at about 40% reduction of labour costs and an
An Approach for Designing Robotized Marine container terminals

Chapter 1: Container sector on the move

overall cost reduction (including the capital costs of the automated equipment) of about 15% up to 25%. It shows that although the investment cost of a robotized terminal are (much) higher, the overall costs can be significantly lower. Especially when labour costs keep rising, which is something that can be expected world-wide, the need for automation of control and equipment increases. The benefits of automation and robotization are confirmed by most terminal operators as well as shipping lines.\(^3\) Besides cost benefits, the following reasons for automation can be mentioned:

- Reducing dependence on scarce skilled labour, such as crane drivers and carrier drivers.
- Ability to handle last-minute changes, such as late-arrivals.
- Ability to change operational priorities.
- Improved abilities for tracing and tracking of containers and equipment.
- Low-cost preparation for upcoming operation (housekeeping).
- Automated feedback loop based on monitoring of key parameters.

![Figure 1-9: Northern-West Europe example of expected cost reduction due to automation (Dobner, 2001c).](image)

Automation and robotization are also fostered by the fast developments in computer technology in recent years, in particular in the fields of information networks,

\(^3\) Information results from personal discussions with people from stevedoring companies, shipping lines and associations that represent companies in the maritime container business and port authorities.
decentralised control systems, software and sensor technology. Although from a cost perspective, automation proves beneficial, European Combined Terminals (ECT) in Rotterdam and Container Terminal Altenwerder in Hamburg are still the only two terminals that have an automated transportation system from the quay – where the vessel is loaded or unloaded – to the stack – where the containers are temporarily stored – by means of automatic guided vehicles (AGVs). Besides Rotterdam, there are three terminals in the world that have automated their yard: two by means of RMGs (Rail Mounted Gantries in Thamesport and again Hamburg) and one by OBCs (Overhead Bridge Cranes, Singapore).

The fact that Rotterdam and Hamburg are the only ports that have gone so far with automation is remarkable, because labour costs are high not only in The Netherlands and Germany. We found (Dobner, 2001c) that the average share of labour cost at container terminals in the Hamburg-Le Havre range varies between 48 and 52% of the total operating cost (see also Figure 1-9). However, recent data from the West-coast of the United States show that the share of labour cost is even significantly higher (up to 65%)\(^4\). Although this favours the application of robotized equipment, the power of the unions – especially in the United States – hinders the introduction of extensive technological improvements (Saanen, 2001).

At present, many container terminal extension and new building projects are under development (see overview in Table 1-4). Many of them are considering automation, at least the internal transportation system from stack to quay and vice versa as well as from the operations in the stack yard. The majority of these terminals will replace part of their equipment and redesign the terminal. Apart from that, a number of terminals are developing new sites into robotized terminals, starting from scratch.

1.6 Trends (3): Towards more dedicated terminals

A clear global trend that can be observed, is the one towards dedicated container terminals instead of multi-user terminals, take for instance the newly realised terminals of APM Terminals (serving mainly Maersk vessels) in Rotterdam, in Los Angeles, in Arhus, in Salalah, the planned dedicated terminals of MSC in Bremerhaven, the dedicated terminal of Evergreen in Rotterdam, and the newly

---

\(^4\) Both on the West coast and on the East coast there are labour unions that negotiate for all the ports on their coast. This leads to similar conditions for all ports.
planned terminals of P&O Ports in Antwerp and London. This coincides with the fact that the control of world’s containership fleet is increasingly in the hands of the top 20 carriers. The main reason for the trend towards dedicated terminals is the improved possibility to integrate and control operations in the entire transportation chain, and the increased ability to get a guaranteed service level. Shipping lines with large volumes are capable of operating terminals themselves and adjust the vessel operation and the terminal to each other.

This trend is also fostered by the extended possibilities to electronically send information in advance, so the terminal is already able to pre-stack for the expected vessels. In addition, real-time decisions can be made about delaying vessels, in the case that no berth is available at the planned arrival time. This can result in lower peak loads and a better utilisation of resources. From an aggregate point of view the efficiency of the transportation chain could benefit very well from these measures. Another possibility that is much easier to realise with a network of dedicated terminals, is to stow a vessel in such a way that the discharge operation in the next port is executed much faster.

Terminal developments that are initiated by terminal operators are in some cases joint ventures between traditional terminal operators and shipping lines, e.g. Euromax in Rotterdam (ECT and P&O Nedlloyd), CTA in Hamburg (HHLA and HapagLloyd), and NTB in Bremerhaven (Eurogate and APM Terminals).

These new terminal developments clearly show that the integration of shipping and transhipment operations are providing efficiency benefits that can only be obtained with more interaction between shipping lines and terminal operators than just information exchange. Examples are adjustment of arrival times, sailing scheme based service levels rather than fixed service levels, and flexibility with regard to the load plan.

The question is whether this far-going co-operation will change the requirements to the terminal operations, since they have traditionally been defined by the demands from the shipping line. For instance, an increase of the berth productivity of a terminal might seem to be profitable for both shipping line and terminal, but analysis shows that this mutual benefit is questionable. In the case of similar volumes, the utilisation of the quay and the equipment at the terminal with the higher productivity
is lower than at the terminal with the lower berth productivity. An example is shown in Table 1-2.

Table 1-4: New container terminals in Northern West Europe (Nieuwsblad Transport, 9 March 2001)

<table>
<thead>
<tr>
<th>Port</th>
<th>Terminal/project ID</th>
<th>Ready</th>
<th>Capacity</th>
<th>Total per port</th>
<th>Quaylength</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Le Havre</td>
<td></td>
<td></td>
<td></td>
<td>800,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port 2000-I</td>
<td>2005</td>
<td>400,000</td>
<td></td>
<td>700</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Port 2000-II</td>
<td>2006</td>
<td>400,000</td>
<td></td>
<td>700</td>
<td>N/A</td>
</tr>
<tr>
<td>Dunkerkens</td>
<td></td>
<td></td>
<td></td>
<td>500,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extension</td>
<td>2002</td>
<td>500,000</td>
<td></td>
<td></td>
<td>P A Dunkirk/IFB</td>
</tr>
<tr>
<td>Vlissingen</td>
<td></td>
<td></td>
<td></td>
<td>2,200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Westerschelde CT</td>
<td>2004</td>
<td>400,000</td>
<td></td>
<td>850</td>
<td>Hessenatie</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2005</td>
<td>800,000</td>
<td></td>
<td>850</td>
<td>Hessenatie</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008</td>
<td>1,000,000</td>
<td></td>
<td>850</td>
<td>Hessenatie</td>
</tr>
<tr>
<td>Antwerpen</td>
<td></td>
<td></td>
<td></td>
<td>4,500,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Schede leftshore I</td>
<td>2002</td>
<td>2,400,000</td>
<td></td>
<td>1250</td>
<td>MSC/Hessenatie</td>
</tr>
<tr>
<td></td>
<td>Closure old inner terminals</td>
<td>2002</td>
<td>-1,100,000</td>
<td></td>
<td></td>
<td>Hessenatie</td>
</tr>
<tr>
<td></td>
<td>Schede leftshore II</td>
<td>2003</td>
<td>900,000</td>
<td></td>
<td>1340</td>
<td>Hessenatie</td>
</tr>
<tr>
<td></td>
<td>Schede leftshore III</td>
<td>2006</td>
<td>700,000</td>
<td></td>
<td>1250</td>
<td>P&amp;N Ports</td>
</tr>
<tr>
<td></td>
<td>Schede leftshore IV</td>
<td>2006</td>
<td>1,600,000</td>
<td></td>
<td>1250</td>
<td>Hessenatie/Noordnatie</td>
</tr>
<tr>
<td>Rotterdam</td>
<td></td>
<td></td>
<td></td>
<td>6,800,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delta Dedicated West</td>
<td>2000</td>
<td>2000</td>
<td>1,500,000</td>
<td></td>
<td>1470</td>
<td>ECT</td>
</tr>
<tr>
<td>Maersk-Delta I</td>
<td>2000</td>
<td>2000</td>
<td>1,000,000</td>
<td></td>
<td>900</td>
<td>Maersk/ECT</td>
</tr>
<tr>
<td>Maersk-Delta II</td>
<td>2001</td>
<td>2001</td>
<td>500,000</td>
<td></td>
<td>350</td>
<td>Maersk/ECT</td>
</tr>
<tr>
<td>Closure ECT capacity</td>
<td>2000</td>
<td>2000</td>
<td>-1,000,000</td>
<td></td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>Euromax</td>
<td>2003</td>
<td>2003</td>
<td>750,000</td>
<td></td>
<td>800</td>
<td>P&amp;O Nedlloyd/ECT</td>
</tr>
<tr>
<td></td>
<td>2005</td>
<td>2005</td>
<td>750,000</td>
<td></td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>Maasvlakte II</td>
<td>2009</td>
<td>2009</td>
<td>1,500,000</td>
<td></td>
<td>1500</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>2011</td>
<td>1,500,000</td>
<td></td>
<td>1500</td>
<td>N/A</td>
</tr>
<tr>
<td>Waalhaven</td>
<td>2001</td>
<td>2001</td>
<td>300,000</td>
<td></td>
<td></td>
<td>Hanno</td>
</tr>
<tr>
<td>Amsterdam</td>
<td></td>
<td></td>
<td></td>
<td>1,800,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>America port</td>
<td>2001</td>
<td>2001</td>
<td>800,000</td>
<td></td>
<td>700</td>
<td>Ceres Paragon</td>
</tr>
<tr>
<td>Africa port</td>
<td>2003</td>
<td>2003</td>
<td>1,000,000</td>
<td></td>
<td>700</td>
<td>Ceres Paragon</td>
</tr>
<tr>
<td>Bremerhaven</td>
<td></td>
<td></td>
<td></td>
<td>1,200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTIII A</td>
<td>2003</td>
<td>2003</td>
<td>350,000</td>
<td></td>
<td>340</td>
<td>Eurogate (BLG)</td>
</tr>
<tr>
<td>CTIV</td>
<td>2006</td>
<td>2006</td>
<td>850,000</td>
<td></td>
<td></td>
<td>800</td>
</tr>
<tr>
<td>Hamburg</td>
<td></td>
<td></td>
<td></td>
<td>3,400,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Burchardkai</td>
<td>2000</td>
<td>2000</td>
<td>800,000</td>
<td></td>
<td>370</td>
<td>HHLA</td>
</tr>
<tr>
<td>Urkai closure</td>
<td>2001</td>
<td>2001</td>
<td>-500,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Altenwerder</td>
<td>2001</td>
<td>2001</td>
<td>1,100,000</td>
<td></td>
<td>700</td>
<td>HHLA</td>
</tr>
<tr>
<td></td>
<td>2003</td>
<td>2003</td>
<td>800,000</td>
<td></td>
<td>700</td>
<td>HHLA</td>
</tr>
<tr>
<td>Giesenwerder</td>
<td>2002</td>
<td>2002</td>
<td>1,200,000</td>
<td></td>
<td>1100</td>
<td>Eurogate (Eurokai)</td>
</tr>
<tr>
<td>Willemschaven</td>
<td></td>
<td></td>
<td></td>
<td>500,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jadeport</td>
<td>2007</td>
<td>2007</td>
<td>500,000</td>
<td></td>
<td>1700</td>
<td>Eurogate</td>
</tr>
<tr>
<td>Felixtowe</td>
<td></td>
<td></td>
<td></td>
<td>1,700,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trinity IV</td>
<td>2002</td>
<td>2002</td>
<td>400,000</td>
<td></td>
<td>270</td>
<td>Hutchinson</td>
</tr>
<tr>
<td>Bathside terminal</td>
<td>2005</td>
<td>2005</td>
<td>1,000,000</td>
<td></td>
<td>1400</td>
<td>Hutchinson</td>
</tr>
<tr>
<td>Efficiency increase</td>
<td>2000</td>
<td>2000</td>
<td>300,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thames harbour</td>
<td></td>
<td></td>
<td></td>
<td>1,000,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New terminal</td>
<td>2004</td>
<td>2004</td>
<td>500,000</td>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>2007</td>
<td>2007</td>
<td>500,000</td>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td>Thamesport</td>
<td></td>
<td></td>
<td></td>
<td>570,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase III</td>
<td>2006</td>
<td>2006</td>
<td>450,000</td>
<td></td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>Efficiency increase</td>
<td>2001</td>
<td>2001</td>
<td>120,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Southampton</td>
<td></td>
<td></td>
<td></td>
<td>1,300,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency increase</td>
<td>2002</td>
<td>2002</td>
<td>300,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dibden Bay I</td>
<td>2005</td>
<td>2005</td>
<td>500,000</td>
<td></td>
<td></td>
<td>350</td>
</tr>
<tr>
<td>Dibden Bay II</td>
<td>2007</td>
<td>2007</td>
<td>500,000</td>
<td></td>
<td></td>
<td>350</td>
</tr>
<tr>
<td>Tilbury</td>
<td></td>
<td></td>
<td></td>
<td>300,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>River Berth II</td>
<td>2002</td>
<td>2002</td>
<td>300,000</td>
<td></td>
<td>300</td>
<td>P&amp;O ports</td>
</tr>
<tr>
<td>Liverpool</td>
<td></td>
<td></td>
<td></td>
<td>350,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal extension</td>
<td>2000</td>
<td>2000</td>
<td>350,000</td>
<td></td>
<td></td>
<td>MDHC</td>
</tr>
</tbody>
</table>

Here, lower berth productivity leads to a better equipment utilisation, but also to a lower vessel service. The vessel waiting times give an idea about the reverse side of the medal: while the terminal equipment may be better utilised, the service towards the shipping lines has decreased. The final answer will be a trade-off between the
cost of an elongated vessel turn-around time versus lower handling cost (per move) under the constraints of the time limits set by the world-wide vessel sailing schemes. In the case of dedicated terminals, this integral comparison, taking cost and performance into consideration is easier made from a single player’s point of view, representing both the terminal operator and the shipping line, than in the case of a multi-player’s situation (e.g. a multi-user terminal).

1.7 Design challenges with regard to robotized terminals
In the previous sections, an overview was given of recent developments in the area of maritime container handling. Summarising, we can say that container terminal operators are confronted with increasing service requirements (especially from shipping lines), and with increasing labour costs. Moreover, the throughput volumes keep rising with a steady rate, which also puts pressure on the terminal operation. Finally, a decreasing availability of land and an increasing burden of environmental regulations stress the need for reconsidering the terminal concept as a whole.

In order to enable terminal operators to meet these changing demands, hard and software suppliers develop new technologies to improve performance and decrease labour costs by means of automation, especially fostered by the increasing possibilities that information and communication technology (ICT) creates. The challenge, however, is to match the increasing service demands from customers on the one hand, and available technology on the other hand. Both have to be seen from a cost point of view as well.

For terminal operators, the integration of automated equipment into the operation has hardly been exploited yet, which can be explained by the relative novelty of the technology, but also by the nature of the sector, which can be characterised as risk-averse. Changing from mainly manual albeit mechanised operations to partly or fully robotized and automated operations is a rather large step. Besides, robotization of equipment also requires redesign and automation of operating tasks, which requires adjustments to the terminal operating system (TOS).

---

5 With hardware we only mean the equipment and facilities at a terminal in this context. The software also includes the computer hardware the software runs on.
Moreover, it is not only a matter of implementing robotized equipment and automated operating tasks into the TOS: the way these processes have to be organised and managed and the way these processes interact are likely to change or have to be changed compared to the manual situation. This also has consequences for the design process of a robotized terminal. Beforehand, the design process of fully automated processes in which a mixture of robotized equipment is used, seems far more complex than a system with only manually operated equipment. Especially the co-operation between hardware and software design and the co-ordination by means of software between different types of robotized equipment in a very dynamic and unpredictable environment are challenges that go along with the present changes of marine container terminals and development of new sites.

There are three main reasons to analyse the design process in more detail. The first and main reason is that the properties of robotized terminals are very different from those of conventional terminals as already argued above. The fact that the control of robotized equipment should be automated means that one of the disciplines involved, namely software engineering, takes on a more prominent role. Whereas traditionally the civil and mechanical engineering were the two most important disciplines involved, software engineering becomes as important when designing automated terminals. Not only the integration of new disciplines is difficult – because of reasons like differences regarding background, culture, procedures, vocabulary, and so forth – also the processes related to the software engineering are different from those of the other disciplines involved. Software testing and software integration – as there does not exist a common-of-the-shelf package for robotized container terminals - are additional, time-consuming tasks that take far less time in conventional terminals. Furthermore, the fact that the job execution of robotized equipment has to be co-ordinated by automated control functions means that interaction between equipment design and software design is a major point, whereas in a conventional terminal, the interaction is limited to the planning and scheduling of operations.

The second reason results the fact that the processes at a robotized, automated terminal are significantly different from those at a conventional terminal. Principally, the handling system should be different to enable robotization, but also the increasing demands, especially regarding the waterside vessel handling capacity,
requires a major improvement of the supplying systems, i.e. the yard handling system and the horizontal transportation system. The processes at a robotized terminal require extensive logistic control functions. Therefore, the terminal operating system (TOS), fulfilling the logistic control functions, is an indispensable component of a robotized container terminal.

The third reason is that the expected demands in terms of performance require a breakthrough in the design of the quay crane, the transportation system, and the yard handling system. This means that we cannot longer build upon the basic performance values of the currently used equipment; the system that will be designed will partly be of a non-existent character. This requires the ability to cope with non-existent equipment and non-existent control rules.

As said in the previous section, large-scale automation is relatively new to container terminal operators and since they can be characterised as risk-averse, the choice for automation should be founded on a thorough comparison between state-of-the-art conventional terminals and robotized terminals. In order to convince the operators that automation brings increased performance as well as lower cost, all aspects have to be analysed in detail. Furthermore, we found (Dobner et al., 2002b) that in order to operate robotized terminals in an efficient way, the logistic concepts should be different from those used at conventional terminals, which means that not only a comparison has to be made, but also new container terminal operating concepts have to be designed.

Summarising, we face a situation where the problem owner, i.e. the terminal operator, is unsure about the best way to fulfil the requirements regarding increasing service requirements, and at the same time accomplishing better levels of efficiency facing more strict environmental circumstances. In addition, there are multiple courses of action, which are unequally efficient or effective. The risks at stake are high – as typical investments in terminal development range from 200 million US$ to 500 million US$. Furthermore, taking the safe road towards manually operated terminal may prove to be unwise: with a share in the total operating cost of 50 to 70%, a likely increase of labour cost immediately affects operating margins.

The decision-making concerning the best course of action takes place in a context consisting of factors that affect the conceptualised and specified alternative courses of action. An effective course of action is defined as leading to the highest possible
valuation of a goal set with given means. An efficient course of action reaches a
certain goal set without using more means than necessary (Sol, 1982).

Therefore, container terminal operators are facing a problem in finding ways to meet
the increased requirements concerning vessel-handling productivities. This problem
consists of three main elements. The first element is the logistical concept of the
robotized container terminal, which is likely to differ from conventional terminals.
The second element (see Dobner et al., 2002b; Verbraeck et al, 1998; Saanen et al,
2000; Saanen, 2002) is the need for an integrated, multi-disciplinary design approach
in order to develop and implement these new logistical concepts successfully.
Thirdly, in order to convince terminal operators to take the hurdle and invest in
automation, a thorough comparison has to be made between conventional terminal
concepts and robotized concepts. In this thesis, all three problem-elements are
addressed.

Given this research motivation, we need to analyse the actual state-of-the-art first, in
order to specify the problem situation. Based on these findings, we can propose a
solution. This solution will then be applied and verified in several cases. This
approach can be categorised as inductive-hypothetic. We choose the inductive-
hypothetic research strategy for a number of reasons (Churchman, 1971; Bosman,
1977; Sol, 1982). First, because of the problem can be classified as an ill-structured
problem, since:

- There are multiple decision-makers and other people involved from multiple
disciplines.
- There are many solutions and none of them is perfect.
- The qualities of the alternative solutions are unknown.

Since the inductive-hypothetic research strategy departs from the principle of
incremental change based on observations and since the current design philosophy
and process cannot easily be changed largely, this strategy seems most suitable for
the kind of problems we are facing. Furthermore, in recent years we carried out a
number of cases (Dobner et al., 2000a, 2001a, 2001b, 2001c, 2002a, 2002b, 2003a,
2003b, 2004a; Verbraeck et al., 1998; Connekt, 2001, 2002) based on which we can
create a rich picture of the current situation.
1.8 Outline of the thesis

The outline of the thesis is as follows. In chapter 2, the experiences from earlier cases and the background resulting from literature research are described. Based on these findings, a framework for an approach for designing robotized container terminals is proposed. This framework, containing the essential properties of such an approach, is described in chapter 3. This chapter forms the overture to the detailed description of the approach, which will be described in detail in the chapters 4-6. In chapters 7 and 8, the application of our proposed approach is discussed and evaluated in two cases. In chapter 9, we reflect on this research, draw conclusions, and suggest recommendations for further research.
2 Review of Container Terminal Design Practices

In this chapter we take a close look at the container terminal as such. Why is it there, how is performance measured, what are typical handling systems, and how does a typical container terminal design process look like? We describe our experiences from the inductive cases, and discuss our research objectives, the research boundaries and research approach.

2.1 Case experience and literature review

During the last 5 years, we performed a number of projects at marine container terminals all over the world, from the United States, Northern-West Europe to Southern-East Asia (Dobner et al., 2000a, 200b, 2001a, 2001b, 2001c, 2002a, and 2002b, 2003a, 2003b, 2004a). These projects have provided us with a lot of case material, which forms the basis for this thesis. Furthermore, we have investigated the design and development process of the Delta Sealand terminal of ECT during the period from 1982 to 1992 by means of interviews (see appendix 10.2), and by analysis of the original system (see Saanen and the Waal, 2001).

Besides, we have performed literature research with regard to the design of marine container terminals, a field where many research has been done already. The research varies from concrete studies at a particular site (see Ballis and Abacoumkin, 1996, Merkuryev et al, 1998), to detailed considerations concerning specific optimisation issues, for instance yard operating strategies (see Kim, 1997, 2002; De Castilho and Daganzo, 1993), equipment assignment strategies (see Steenken, 1992, 1993; Van der Meer, 2002) or berth allocation strategies (see Edmond, 1978; Imai et al., 1997, 2001; Kim and Moon, 2003), and to the development of models to compare various handling systems (see Bruzzone, 1999; Yun and Choi, 1999; Liu, 2001). In this chapter, we will try to capture the existing knowledge on container terminals, and add our experiences with regard to automated terminals.

2.2 Functional description of a container terminal

2.2.1 Functions of a container terminal

The container terminal, subject of this research, can be considered a (complex) system. In our definition of a system, we follow Webster's Third New International Dictionary (1971). It defines a system as "an aggregation or assemblage of objects joined in
"some regular interaction or interdependence". The main attributes of a system are: its functions (expressed in a system’s activities), the main performance requirements to the system, and the resources the system consists of. These attributes will be discussed in the next sections.

A container terminal fulfils two main functions within the intercontinental transportation chain: transhipment from mode to mode as well as temporary storage of containers (see Figure 2-1). The external added value is provided by the speed at which sea vessels are handled, and the decoupling of sea vessel arrivals and the arrivals of continental transportation modes, such as trucks, trains, short-sea vessels and barges (see Figure 2-2). The storage functionality should not be underestimated: container terminals provide a relatively inexpensive, secure, easily accessible location, from which just-in-time delivery of containers can take place. The reasons that the storage functionality of a marine container terminal is essential are summarised by Zijderveld (1995):

1. Direct transhipment would make the processes at the terminal too complex; in case of transhipment from a deepsea vessel onto trucks, this would result in a complex controlling of all individual trucks, to make sure that they arrive in the right order at the right time to get their container.

2. Direct transhipment would yield a complex terminal design for those terminals that include more than two modes of transport: All modes of transport that are
handled would come very close to each other, which would immediately cause difficulties for terminals that include barges, deepsea vessels, trucks and trains.

3. Direct transhipment would require the simultaneous presence of both means of transport between which load units are transhipped. Especially direct transhipment of load units between deepsea vessels and trains or even between vessels is virtually impossible due to the great diversity of destinations of load units, the strict loading sequences of trains and vessels, and the length of trains.

4. The owners of containers do not always need their cargo immediately after the arrival of a container. Also, customs demands (fulfilment of duties) and financial requirements will cause the need to store the container(s) at the container terminal. As research shows (Rakt, 2002), some containers stay more than 6 months at the terminal.

Besides those two primary functions of a marine container terminal, the terminal also fulfils the following secondary functions (Hekman, 2001):

- Consolidation function: Containerised cargoes may be shipped on an FCL (Full Container Load) or LCL (Less than Container Load) basis, the latter being cargoes with different origin/destination points consolidated into containers. The consolidation function in a Container Freight Station (CFS) is performed by some container terminals, but it is not an indispensable requirement.

- Back-up function: This function supports the performance of the other three functions (transhipment, storage, and consolidation). For example, a maintenance shop for equipment and facilities, and container handling equipment inspection and repair.

- Depot function for empty (MT) containers and/or shipping line owned road chassis.

2.2.2 Processes at a container terminal

When we take a closer look at a container terminal, we can discern the following processes that take place (see Figure 2-2):

- Berthing of vessels at the quay.
The discharge process of the vessel, and the handling of containers from the vessel to an internal transportation mode (terminal tractors with chassis, straddle carriers, reach stackers, automated guided vehicles).

- Transport of containers to the stack.
- Unloading of the internal transportation mode, handling of containers to the stack.
- Storage of containers.
- Internal shifting of containers in the stack (housekeeping).
- Transhipment of containers from the stack to a landside internal transportation mode.
- Transport to the transhipment point of the continental transportation mode (barge, rail, trucks, short-sea, and feeders).
- Handling to the continental transportation mode.

This sequence of processes also takes place in the opposite direction.

Figure 2-2: The container terminal: interlinking various transportation modes (Rijsenbrij, 1998)

The emphasis on the functions of a container terminal and the processes taking place differs among the various types of containers terminals, as will be discussed in section 2.5.
2.3 Hardware at a container terminal

2.3.1 Introduction
In the previous sections, the processes at a container terminal have been described. In order to have an idea what kind of equipment is involved in those operations, we present here a short overview of the equipment types and their properties. The equipment is not only interesting as background information, but also because equipment selection (amount and equipment characteristics) is one of the main issues in container terminal design.

2.3.2 Transhipment from vessel to quay and vice versa
The use of equipment differs among the container terminals all over the world. The main equipment is the quay crane, which unloads and loads the vessel. There are two types of quay cranes: rail gantry cranes and mobile harbour cranes (see illustrations in Figure 2-3). The rail gantry crane is less flexible than its rubber tyred mobile opponent, because of its limited move ability. On the other hand, because of its rigidity, it performs better than mobile cranes because of the higher hoisting and trolley speed of gantry cranes and its proper load control. These higher speeds can be accomplished because mobile cranes have to turn with the whole crane to transship the container from the vessel to the quay, whereas a container gantry crane or ship-to-shore crane only needs trolley travel.

Figure 2-3: A quay gantry crane and a mobile harbour crane

Moreover, the capabilities of the two types of cranes in terms of the types of vessels that can be handled differ. The largest vessel a mobile can handle is 16 to 17
containers wide on deck. State-of-the-art gantry cranes can handle vessels up to 22 containers wide on deck. In practice, mobile cranes are used on vessels up to 13 containers wide on deck.

The trend towards larger vessels has to be followed by larger cranes and faster cranes, hence if all other things are equal, the cycle time of the cranes increases. In the figure below, the relationship between vessel width and crane cycle time of different quay cranes is depicted.

![Figure 2-4: Relationship between operational productivity and vessel size; productivities based on state-of-the-art post-Panamax quay crane. Vessels as depicted in Figure 1-6.](image)

So, although the call sizes are increasing, the productivity rate with which the quay cranes are able to operate is decreasing due to longer travel distances for the cranes as a result of the larger vessels. So, the challenge is to decrease the gap between the attainable operational productivity (as depicted in Figure 2-4) and the achieved gross productivities under operational conditions. One way that terminal operators tend to follow, is to install dual trolley quay cranes, where the vessel is handled by one trolley, which puts the container at a platform. There, the twistlocks are removed. Then a second trolley – preferably automated – brings the container to the landside interchange point where it is being handed over to the horizontal transportation system. These cranes typically increase achievable productivity with about 15 to 20%.
2.3.3 Internal transportation

The equipment that is used for the transport processes at a terminal varies considerably. Terminal tractors with trailers, straddle carriers, shuttle carriers, multi-trailers, and automatic guided vehicles (see Figure 2-5) are examples of the equipment that is used to transport containers from the quay to the stack.

![Figure 2-5: Left: an automated guided vehicle (courtesy of ECT, 2001); right: a straddle carrier.](image)

2.3.4 Yard handling

To put containers in the stack, move containers within the stack and to get containers out of it, different types of stacking cranes are used. The most common types of yard handling equipment are the Rubber Tyred Gantry (RTG), the Rail Mounted Gantry (RMG), and the Overhead Bridge Crane (OBC), merely known from industrial applications. Straddle carriers are also used as equipment to operate the stack.

![Figure 2-6: Yard handling equipment at a container terminal, a RTG (left), a RMG (ECT, 2001).](image)
2.4 Software at a container terminal

Although at first sight a terminal may mainly consist of hardware, the importance of software should not be underestimated. Although many decisions at conventional terminals are still made by planners, crane operators and equipment drivers, most operations are supported by terminal control software. Whereas software was originally introduced to support administrative processes (e.g., EDI, container information registration), slowly but surely, software is taking over more operational/planning tasks. Nowadays, comprehensive software packages are available, of which Express-Sparsc from Navis and Ships-Space-Trafic of Cosmos are the most widely used. These packages support almost every operation at a container terminal, from creating a stowage plan, to scheduling of transportation equipment, yard planning and the gate process. However, up to now there are no common-off-the-shelf packages available to control an automated terminal, such as the Delta terminal of ECT or the Altenwerder terminal in Hamburg, which has caused these terminals to develop the software themselves. Furthermore, there is hardly any software available for the operational control of a complex of multiple terminals, in between which inter-terminal transport is executed.

The lack of common-off-the-shelf software for automated terminals makes the terminal operators uncertain about the choice for automation, because automated terminals rely heavily on software. A recent investigation (Saanen, 2002a) has shown that most terminals would choose common-of-the-shelf packages whenever available, because they usually contain fewer errors, are less costly to maintain and have been proven in a terminal operation.

2.5 Classification and assessment of container terminals

Container terminals have to meet certain objectives regarding service and efficiency, in order to stay competitive. These requirements may vary depending on the terminal location and the terminal characteristics. In some areas, travelling distances are relatively small (Western Europe, South-East Asia), which stimulates container terminals to distinguish themselves from other terminals by offering unique or high quality service. An example of this is the excellent connection by inland waterway transport of the Rotterdam and Antwerp ports with their hinterland (i.e. Ruhr-area in Germany). Some of the demands regarding terminal service and performance are listed below (Rijsenbrij, 1998):
− Sufficient draught; at least 16.5 m is currently (2004) required for the hub ports.
− Operating 7 days a week, 24 hours a day.
− Providing a total vessel handling operation (time-in-port) within agreed times. For container vessels, this period usually is 24 hours, since service time clearly has a large influence on the vessel’s operating costs.
− Stability in service performance even if a greater number of ships call on the same day or when the landside shows peaks at the same time as the waterside (support of simultaneous waterside and landside operations)
− Safe handling of cargo from the point of view of both men and cargo.
− Environmental control (emission, noise) during handling operations.
− A 100% reliable, real-time control system (for physical handling and information flows) with the on-line presentation of data (preplanning; tracking and tracing etc.) and state-of-the-art EDI services.
− Back-up activities (such as a container freight station, reefer plant, empty container depots, trailer parking, maintenance and repair, et cetera) are at or close to the terminal.
− Auxiliary services for a variety of activities related to the cargo transportation. These may include: fresh water supply for the vessels, services for small ship repairs, et cetera.

Whenever it comes to the functional design of a terminal and the related forecast of productivity and handling costs, it is necessary to analyse the total package of services offered by the terminal. In order to measure the (service) performance of a container terminal, some indices are used for analysis and evaluation. This section discusses the most frequently used indicators (Watanabe, 2001; Ugvlig, 2001) and ratios, and gives some rules of thumb. We distinguish between design indicators, classifying the type of terminal, and performance indicators, aimed at assessing the service level that a terminal provides.

In order to evaluate the functionality and service a terminal provides, performance indicators have to be defined, which are linked to the objectives of a terminal (Saanen, 1996). We apply an outside-in approach here, starting with the functionality provided to the customers of the terminal. First, we mention a number of (design)
indicators that determine the terminal in terms of size and type of operation. Then, we mention the four most important performance indicators on the most aggregated level. Subsequently, we zoom in to performance indicators that allow us to assess the handling system in more detail. Finally, we define a number of indicators to assess the operational costs. In addition to the individual description of the indicators, we also show some interdependencies between the indicators.

2.5.1 Design indicators to classify various terminal types

A terminal can be classified by a number of factors. First, there is the volume, which is determined by the location and/or the local economic conditions. The volume is usually expressed as the number of waterside moves on sea going vessels (see Equation 1). Then, there are a number of site-specific conditions that could limit the choice of handling system, such as the soil conditions and the shape of the land (width and depth). A determining factor is the percentage of transhipment moves, i.e., the moves that arrive over the waterside and depart over the waterside as well. True transhipment terminals, such as the terminals in Singapore, have transhipment percentages of over 80% (Connekt, 2001), whereas transhipment percentages of typical origin/destination terminals – for instance at ECT in Rotterdam are at a rate of 25% to 30%. The transhipment factor (see Equation 2), together with the TEU factor – representing the ratio between TEU (twenty feet equivalent units) containers and 40 feet (FEU) containers according to Equation 5 – the throughput (see Equation 1), and the dwell time, determine the storage demand of a terminal. Since a transhipment move leads to one visit to the stack and two quay moves (the same container is counted twice as quay move, but once in the stack), we define the number of stack visits (measured in TEU as this is a storage unit) to determine the storage demand. Here also moves that enter via landside and exit via the landside (so called domestic moves) have to be counted (see Equation 6).

Equation 1: Terminal throughput

\[
\text{Terminal throughput (containers)} = \text{All productive moves made by QCs on sea going vessels.}
\]

Equation 2: Transhipment factor

\[
\text{Transhipment factor} = \frac{\text{All moves that origin from sea going vessels that leave with sea going vessels}}{\text{Terminal throughput}}
\]
Terminal turnover = \frac{\text{Terminal throughput (moves)}}{\text{Average number of containers in the terminal}}

**Equation 4: Dwell time**

\[ \text{Dwell time (days)} = \frac{365}{\text{Turnover}} \]

**Equation 5: TEU factor**

\[ \text{TEU factor} = \text{Percentage 40 feet containers} + 1 \]

**Equation 6: Stack visits**

Number of stack visits (TEU) = \text{Terminal throughput} \times (1 - \frac{1}{2} \times \text{transhipment factor}) + \frac{1}{2} \times \text{domestic moves} \times \text{TEU factor}

**Equation 7: Storage peak factor**

\[ \text{Peak factor} = \frac{\text{Maximum number of TEU in yard}}{\text{Average number of TEU in yard}} \]

**Equation 8: Average storage demand (measured in TEU)**

\[ \text{Average storage demand (TEU)} = \frac{\text{Stack visits} \times \text{Dwell time}}{365} \]

**Equation 9: Maximum storage demand (measured in TEU)**

\[ \text{Maximum storage demand (TEU)} = \text{Average Storage Demand} \times \text{Peak factor} \]

**Equation 10: Required storage TEU groundslots**

\[ \text{Required TEU groundslots (TGS)} = \frac{\text{Maximum storage demand}}{\text{Maximum stacking height} \times \text{Maximum filling rate in peak}} \]

Other factors that have to be mentioned are the type of vessels calling the terminal; the bigger the vessels, the larger the draft and the bigger the cranes; the local economical situation, determining e.g. the labour cost and the availability of labour, and the scarcity of land; the scarcer the available land is, the denser the storage area at a terminal has to be.

The following indicators are relevant for classification of a container terminal:

- **Annual container handling capacity (TEU/year)**

The annual handling capacity is the typical indicator for the transfer function of terminals. It has been standardised in TEU, although it is a combination of handling
capacity (measured in moves), and storage capacity (measured in TEU). The quay length, the waterside handling capacity, the storage capacity, the landside handling capacity (gate, rail, and barge) including connecting infrastructure, and the available handling equipment (for transportation and yard) determine this capacity.

- **Ratio of transhipped containers (μ)**

The parameter $\mu$ represents the ratio between transhipped containers and import/export containers and assesses the characteristics of the container terminal. The higher this ratio is, the more containers that enter the terminal via deep-sea or feeder vessels, also leave the terminal via these modes. All other containers are counted as import-export containers. See Figure 2-7 and Figure 2-8.

Container terminals can be divided into three categories: Hub centre, OD (origin-destination) or local ports. For a value of $\mu \leq 0.5$, facilities and equipment for landside operations such as the gate, interchange areas and yard handling equipment are the most important considerations for planners. These terminals are mostly categorised as OD type (Watanabe, 2001). When $\mu \geq 0.5$, the container terminal is usually classified as a hub centre type (or transhipment type), and facilities and equipment for waterside operations are the most important for planning purposes. Local ports (throughput < 250,000 TEU annually) mostly handle only smaller vessels and have a similar transhipment ratio as OD terminals.

The transhipment percentage is, besides the terminal throughput and the dwell time, the key factor to determine the required stack capacity. For the same terminal throughput (in waterside moves), a 100% transhipment terminal requires half the stack capacity of a 100% OD terminal. The latter notion is crucial here because the volume a terminal handles is only measured by the number of containers that are transhipped over the quay; usually the landside moves are not counted. One remark here has to be made with regard to containers that are moved to the hinterland by means of barges. This volume is sometimes handled over the deepsea quay, and sometimes over a separate facility. In the first case, it means that additional quay volume is generated, which is not counted as throughput volume over the quay wall, although it increases berth utilisation and quay crane utilisation. A better approach would be to count these moves into the transhipment percentage in the case that these containers are moved over the deepsea quay wall. In the second case, where
there is a separate barge handling facility, this effect does not occur and therefore the barge volume can be considered as import/export flow.

![Diagram of container terminal design](image)

**Figure 2-7:** Transhipment factor $\mu$ in relation to container flow; in case of a separate barge facility.

![Diagram of container terminal design](image)

**Figure 2-8:** Transhipment factor $\mu$ in relation to container flow in case barges are handled at deepsea quay.

- **Storage capacity (TEU)**

The storage capacity is the second most important index as it determines the character of the storage function of container terminals. Characteristics of the
storage capacity are for instance, the number of TEU ground slots (TGS), the maximum number of container stacking tiers, and the maximum operational filling rate of the stack. The storage capacity is measured in a standardised volume unit, the TEU.

![Figure 2-9: Relationship between transhipment percentage and required stack capacity (Hekman, 2001).](image)

2.5.2 Performance indicators from a customer point of view

- **Maximum vessel size**
  For shipping lines, the capability of a terminal to handle vessels (draught and vessel handling capacity) determines, among other factors, whether they will call at a terminal. With an increasing size of vessels, the need for draught and larger cranes (lifting height and outreach) changes (see also Figure 1-6).

- **Berth productivity or vessel service time**
  This indicator is a measure for the speed with which a vessel is handled. The required time window in which the vessel has to be handled depends on the vessel's sailing scheme and type of call. The time-in-port is especially important for deep-sea vessels, because the excess of an agreed time window may result in a disturbance in the sailing scheme.

In some cases (Ugvlig, 2001), shipping lines define quay crane availability as a service indicator. Although from an operational point of view the number of cranes is less important than the productivity achieved on a vessel, the number of cranes available for a vessel service is historically an important factor.
Landside service times
At the landside of the terminal, trucks, barges, and trains are serviced. As with the
deep-sea vessels their service time is an indicator of terminal performance. An
interesting difference exists here between merchant and carrier haulage. In the case
of merchant haulage, the intercontinental and continental transportation chains are
not linked: the continental part is performed by another enterprise than the carrier
(shipping line). This means that another enterprise, without any contractual relation
with the terminal, picks up or delivers a container. In the case of carrier haulage, the
shipping line also takes care of the continental part (for instance Maersk-Sealand and
P&O Nedlloyd perform a lot of carrier haulage), which means that there is a greater
possibility to adjust land- and waterside operations to each other. Moreover, in case
of carrier haulage, terminal operators have more incentives to influence the
customer, because for both operations a contractual relationship exists.

Last minute changes
In a time that logistics get increasingly real-time, the need for flexibility increases. For
terminals, this means that they have to cope with last minute changes in vessel load
lists and with late container arrivals at the terminal. The easier a terminal can handle
late changes, the more service it can provide to its customers. This indicator is of
increasing performance and is used to compete against cost (Chiu, 2002).

Price per move
The price per move is together with the berth productivity the most important
indicator, since they determine the attractiveness of a port. Of course, the
productivity and cost per move have to be seen as a mutual trade-off. The other side
of the price per move is the cost per move, including all costs made on the terminal.
The prices that are agreed upon with shipping lines differ heavily between contracts
and depend on the specific service agreement and the yearly volume.

Price per storage day
The average dwell time – the number of days a container stays in the yard – of most
terminals varies between three to six days (Hekman, 2001). However, in a number of
terminals the space in the yard is limited, so they preferably keep the dwell time low
by charging a higher cost per storage day. Storage day pricing heavily differs among
terminals; it is determined by regional competition and scarcity of land in that area.
2.5.3 Performance indicators of terminal utilisation and productivity

The following indicators are used to assess the level of efficiency of a terminal:

- **Annual container handling capability per area unit (TEU/ha)**
  
  Based on the rated annual container handling capability it is a convenient index for planning, analysing or benchmarking container terminals. As a rule of thumb, the maximum annual container handling capability per total area of the container terminal is 23,000 TEU per ha/year for OD-type terminals and 50,000 TEU per ha/year for hub-centre terminals (Watanabe, 2001). The higher value of the latter can be explained by the lower dwell time for transhipment containers, the lower area requirements (less visits per waterside move) of transhipment containers (see section 2.5.1), and less area for secondary activities.

- **Turnover in the yard or dwell time**
  
  The turnover of the spaces in the yard has a reciprocal relationship with the dwell time. To find the turnover of TEU spaces in the yard, the annual container handling capability or actual container traffic (TEU/year) is divided by storage capacity (TEU).

  The dwell time can be measured by measuring the time every container stayed at the terminal, according to the following formula:

  
  \[
  Dwelltime = \frac{1}{n} \sum_{i=1}^{n} (TimeofArrival_i - TimeofDeparture_i)
  \]

  As dwell times can vary between several hours until months, it can only be measured over longer periods of time. However, dwell time should not be mixed with the notion of the age of containers, which is basically a snapshot of all containers at the terminal, and their actual average duration of stay at the terminal. As the dwell time variation is large in most terminals and the weight of containers that stay long at the terminal is proportionally high in such a snapshot, the average age is in most terminals higher than the average dwell time, although one would expect that the age is lower: in case of no variation in dwell time, the average age would be \(\frac{1}{2}\)x the dwell time.

  Usually dwell times vary for different types of ports (hub-centre, OD, local). The average dwell time for a hub centre container terminal type is around three to five
days, while the average dwell time for an OD type of container terminal can vary between 5 and 15 days (Drewry, 1998).

![Comparison: waiting times for berth assignment](image)

![Distribution of volume over the week](image)

Figure 2-10: Two vessel call patterns (upper figure) influencing the waiting time before berthing (lower figure) (Dobner et al., 2001e)

- **Annual container handling capability per metre quay (TEU/m)**

The required length of quay wall is usually determined by the maximum size of calling vessels as well as their calling pattern, not by operational requirements of the terminal such as storage capacity, annual handling capability and so on. The annual
container handling capability per total length of quay wall varies between 150 to 2000 TEU per m/year. Because the quay wall is one of the main cost factors of a terminal, the throughput in relation to the quay wall length is an indicator to check the terminal’s efficiency. However, the terminal operator has limited possibilities to influence the calling patterns of vessels, especially at a multi-user terminal. Hence, the throughput per meter quay depends on the co-operation between the shipping lines – determining the vessel call pattern – and the terminal operator – determining the berth and terminal handling facilities.

An example is shown in Figure 2-10 (Dobner et al., 2001e). In the upper diagram, the number of lifts per day is depicted for two call patterns. These lifts are distributed over a number of vessel types. Calling pattern two (orange) has higher peaks on Tuesday, Thursday, and Friday; the total yearly volume is equal. In the lower diagram, the effect of the call pattern on the waiting time before berthing is depicted. This waiting time is an indicator for the availability of the quay at the time a vessel arrives. As can be seen in the lower diagram the percentage waiting vessels in the second call pattern scenario is significantly higher than in the upper call pattern scenario. Depending on the requirements of the shipping line, the terminal operator would have to elongate its quay and install more quay cranes to reach the same service level in the second scenario, although the transhipment volume remains equal.

2.5.4 Performance indicators of productivity of container handling equipment

- **Productivity of quay cranes (containers / crane hour)**

The productivity of a quay crane is defined as the number of containers handled by the crane per hour during a vessel operation. This is one of the most important indicators for analysis and evaluation of the productivity of existing container terminals, as well as for planning of new container terminals, because quay cranes are expensive and the quayside determines a significant percentage of the total terminal costs. In addition to the technical specifications of the crane, there are some operational measures that can improve the performance of the crane: twin-lift, tandem-lift and dual cycling.

- Twin lifting means that two (20 feet) containers are moved in a single crane movement; it doubles the productivity of those crane cycles.
- Tandem lift means that two forty feet containers are moved side by side.
Dual cycling means that in one (elongated) crane cycle a container is discharged and loaded. Although this seems very promising, the operational practice limits the possibilities for dual cycling due to stowage plans and physical restrictions.

- **Annual number of containers handled by a quay crane (containers/year)**
  This index is mainly used for macroscopic analysis, to assess whether or not the number of quayside container cranes in a port is appropriate to the level of throughput. A benchmark for container terminals around the world is to acquire an extra crane when the average number of containers handled per crane year exceeds 70,000 containers, of course depending on the vessel call pattern. The newest generation of container vessels requires higher quay cranes with a longer outreach. Although this leads, all other things being equal, to longer individual crane cycle times, the larger cranes have a higher annual handling capacity due to the reduction of crane movements, hatch cover handling and other operational disturbances. The average number of containers handled per crane per year is therefore higher for hub-centre terminals, which (usually) serve larger container vessels. For hub-centre terminals, this number can be >100,000 containers per crane per year (Dobner et al., 2002b).

- **Productivity of container handling equipment (moves/equipment hour)**
  The number of containers (or TEU) handled by the equipment per hour is the productivity of the container handling equipment. This is one of the most important indicators for analysis and evaluation of the productivity of existing terminals as well as for dynamic planning of new container terminals. Exact determination of handling equipment productivity is difficult because it is subject to many variables, such as waiting times for connecting equipment, operating strategies, conditions in the yard, the area, the layout, operator skills, equipment characteristics et cetera. In general, for terminal equipment, four different kinds of productivity can be discerned (see Figure 2-11):
  - Technical equipment productivity: this productivity is determined by physical factors like average travel distance, speed of the machines, et cetera. No interferences from other equipment, manual interventions, or disturbances have been taken into account.
− Operational equipment productivity: this is the productivity of a piece of equipment in an operational cycle. Included are the delays due to the driver, and external influences such as wind, surface conditions, et cetera. Excluded are the influences due to the deliver or pick-up of containers by other types of equipment.

− Net equipment productivity: during the operation, equipment usually has to wait for other equipment to interchange with or because of other operational disturbances. The productivity that results from the number of productive moves divided by the production time is the net equipment productivity. This productivity can only be determined in interaction with other equipment as far as it influences the equipment productivity.

![Figure 2-11: Relation between productivities](image)

<table>
<thead>
<tr>
<th>Quay crane productivity</th>
<th>ecph*</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ) Kinematic quay crane specification (trolley, hoist)</td>
<td>55</td>
</tr>
<tr>
<td>( ) Type of vessel, stowage plan</td>
<td></td>
</tr>
<tr>
<td>Technical productivity</td>
<td>55</td>
</tr>
<tr>
<td>( - ) Sway, crane-driver skills</td>
<td>50</td>
</tr>
<tr>
<td>( - ) Disturbances due to lashing, positioning, twistlock handling</td>
<td></td>
</tr>
<tr>
<td>Operational productivity</td>
<td>50</td>
</tr>
<tr>
<td>( - ) Waiting for waterside transportation system</td>
<td>40</td>
</tr>
<tr>
<td>Net productivity (target / output simulation)</td>
<td>40</td>
</tr>
<tr>
<td>( - ) Break-downs</td>
<td>35</td>
</tr>
<tr>
<td>( - ) Meal-breaks</td>
<td></td>
</tr>
<tr>
<td>( - ) Shift effects</td>
<td></td>
</tr>
<tr>
<td>( - ) Hatch cover handling</td>
<td></td>
</tr>
<tr>
<td>( - ) Bay changes</td>
<td></td>
</tr>
<tr>
<td>Gross productivity (target, net -10%)</td>
<td></td>
</tr>
</tbody>
</table>

*ecph = equipment cycles per hour

Figure 2-11: Relation between productivities

− Gross equipment productivity: besides the regular waiting times that are inherent to an operation, an operation is usually not continuous; meal breaks, quay crane breakdown, et cetera, lead to a lower productivity over the entire time of vessel handling. The gross productivity is measured between the start and end of a vessel operation, which time is normally longer than the
productive equipment time, because it includes all kinds of disturbances. Therefore:

\[
\text{Gross productivity} < \text{Net productivity} < \text{Operational productivity} < \text{Technical productivity}.
\]

- **Density of quayside container cranes (m/crane)**
  This index is the length of quay wall furnished with crane rails per crane. In most terminals, this index is >100 meters. Some terminals in Asia have values that are smaller than 100 meters. A too small index may cause heavy congestion for horizontal transportation equipment needed to support waterside operations on the berth apron (for instance long queues of AGVs). The density of quayside container cranes is usually higher in hub-centre type container terminals than in OD terminals. The reason for this is the higher number of waterside moves per hour per metre quay wall in hub-centre terminals in relation to the required stack size. More (waterside) quay cranes are therefore necessary. When the quay length stays the same, this means the density of quayside container cranes increases.

2.5.5 **Performance indicators for land use efficiency**

- **The area rate of the marshalling yard (%)**
  The ratio of area of the yard to the total area of the container terminal is the area rate of the yard. This index is in the range of 0.5 to 0.7 for terminals with container freight station (CFS) within their boundaries, and 0.6 to 0.8 for terminals without them (Watanabe, 2001). Logically, limiting the amount of other area than the yard saves money. This only applies in case terminal operations are not limited by matters like traffic congestion, caused by the lack of sufficient traffic lanes and manoeuvring space.

- **Yard density (TEU/ha)**
  This index shows the number of TEU per hectare of the yard under operational conditions. That means that considerations such as the maximum filing rate are included. The values vary for different handling systems, in practice from approximately 700 TEU/ha for a straddle carrier (SC 1 over 3) system to 1,500 TEU/ha for an overhead bridge crane (OBC) system (Dobner et al., 2002b).

- **Accessibility of containers stacked in the yard**
  The accessibility of containers is defined as the average number of moves that is necessary to retrieve a specific container in the stack. A high accessibility is represented
by one move (just picking up the right container), while a low accessibility results by
more moves (picking up the right container and the required number of re-handles).
The accessibility represents the relation of the transfer function with the storage
function in the terminal. A good accessibility of stacked containers is important for
the terminal capacity. The better the access to containers, the fewer re-handles that
are necessary and the higher the effective net productivity of the yard handling
equipment. The accessibility depends on the number of tiers (stacking height), the
order in which containers will be retrieved and the quality of the available container
information (load lists) when the container is grounded in the stack. In an optimal
situation the container information is 100%, which enables a stack planning system
to reduce re-handles by assigning accessible positions. In practice, however, this does
not seem to be feasible in the short term (Ugvlig, 2001; Ham, 2001; Dobner et al.,
2001c).

Figure 2-12: Typical distribution of operational cost (Dobner et al., 2001c).

2.5.6 Container terminal operating costs

A major issue concerning the design of container terminals is the cost. Here we have
to discern between the investment costs – the capital that has to be put into the
terminal to acquire the equipment and facilities – and the yearly operating costs.
Although in the long term the operating costs determine whether the terminal is
profitable or not, the importance of the investment cost may not be underestimated.
The investment costs determine the payback period, and to reduce risks, operators tend to use a maximum payback time of 10 to 15 years. Especially robotized equipment – for instance, RMGs – tends to live longer than this period.

The main components of the investment costs are the facilities (quay, paving, buildings) and the equipment. The main components of the operating cost are capital cost, labour cost, maintenance cost, and cost of land (mostly lease).

Table 2-1 gives some estimated container terminal equipment investment costs. The dominance of quay cranes in the overall costs is immediately apparent. When analysing the total cost of equipment of a terminal – for instance a one million TEU terminal with 1,000 meters of berth and 9 quay cranes and 36 straddle carriers –, one arrives in the range of € 75 to € 150 million.

Table 2-1: Estimated container terminal equipment cost (Drewry, 1998; Dobner, 2002b)

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Investment cost (Euro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reach stacker</td>
<td>€ 325,000</td>
</tr>
<tr>
<td>Forklift Truck</td>
<td>€ 250,000</td>
</tr>
<tr>
<td>Quay crane</td>
<td>€ 5,000,000 - € 7,500,000</td>
</tr>
<tr>
<td>Spreader</td>
<td>€ 125,000</td>
</tr>
<tr>
<td>Tractor/Trailer</td>
<td>€ 90,000</td>
</tr>
<tr>
<td>Automated Guided Vehicle</td>
<td>€ 350,000</td>
</tr>
<tr>
<td>Automated rail mounted gantry (6-wide)</td>
<td>€ 1,100,000</td>
</tr>
<tr>
<td>Automated rail mounted gantry (9-wide)</td>
<td>€ 1,600,000</td>
</tr>
<tr>
<td>Straddle Carrier</td>
<td>€ 500,000</td>
</tr>
<tr>
<td>Radio/communication system (total)</td>
<td>€ 300,000 - € 500,000</td>
</tr>
<tr>
<td>Computer hardware and software (total)</td>
<td>€ 4,000,000 - € 20,000,000</td>
</tr>
<tr>
<td>Service vehicle</td>
<td>€ 14,000</td>
</tr>
</tbody>
</table>

The labour cost is most interesting, when looking at automated terminals, since the largest saving comes from the reduction of the workforce. Not surprisingly, labour costs vary considerably from port to port and country to country. Local wage rates, union power, and historical factors all play a part in this. Waterside workers generally earn a considerable amount of overtime and in a modern privately run container terminal, incentive bonus schemes often add between 25% and 50% to the basic pay packet (Drewry, 1998).

The operating benchmark results indicate an average TEU/headcount varying from 900 for a small container terminal to 1,100 for a large container terminal, resulting in average labour cost/TEU from € 19 (small container terminal) to € 15 (large container terminal) (Drewry, 1998). Compared to other studies, these figures are rather low. Dobner et al. (2002b) come to a labour cost/TEU in the range of € 30
and € 38 for a Northwest European terminal. Especially in the United States, the labour costs are relatively high compared to Europe and Asia. An indication for the share in the labour cost of the total operational cost varies between 35% (Far East) towards 50% (Northwest Europe) and 65% (United States, West coast) (Dobner, 2001d; Saanen, 2001c).

2.6 Design of a container terminal

2.6.1 Relevant trends

The following trends with regard to supply chains can be observed (Boyson et al., 1999:12):

− Real-time connectivity among extended-enterprise partners and deployment of fast global information exchange networks within an increasingly open standard operating environment based on Internet protocols.

− A more holistic, system-engineering and integration approach to value-chain management and more precise methods of costing and controlling transaction flows, such as activity-based costing and business process re-engineering.

− An increased use of intermodal transportation for international and domestic shipping.

When we consider these general trends, we observe several similarities with the trends regarding marine container terminals. First, marine container terminals, since they are heavily dependent on the processes that take place outside of the terminal (such as the arrival of vessels, trains and trucks), try to acquire improved information provision by external parties. These types of external information concerns arrival or departure times of vessels, actual weight, port of destination of containers, estimated pick-up times of containers, et cetera. Here, two developments can be observed. On the one hand, terminal operators try to obtain the information earlier. On the other hand, they try to improve the accuracy of the information. However, the success of these efforts has been very limited up to now. Especially at so-called multi-user terminals, the direct influence of a terminal operator (the stevedore) on the water- and landside arrival patterns is limited. Terminal operators that control the intercontinental and continental operations as well – for instance Maersk-Sealand is both terminal operator, shipping line and transportation company – have more opportunities to optimize the transportation chain at a higher level than terminal
operators within a supply chain do, when multiple enterprises are involved. As supply chain integrator, it is possible to adjust the terminal operations and, vessel schedules to one another in order to avoid peaks and, therefore unreliable service times or an inefficient terminal operation.

Secondly, marine container terminals subscribe to the observation that a holistic view is required when designing a complex system that is significantly influenced by external factors. Many studies on container terminals (Visser, 2000; Ottjes et al., 1996; Duinkerken and Ottjes, 2000; Blümel, 1997; Merkuryev et al., 2001; Ezquerra, 2001) focus on the waterside operations of a terminal or even at a specific part of the terminal (e.g. Kim, 1997, 1999, 2002). Although all these studies are valid within their scope, we have found (Saanen and De Waal, 2001a; Dobner et al., 2003a, 2003b) that the interaction between landside operations and waterside operations significantly influences the terminal operation as a whole. Meersmans (2002) stresses the importance of integrated scheduling of yard handling- and transportation equipment, and has concluded that little attention is paid to such an integrated scheduling approach. Furthermore, we found that the choice for a certain internal transportation system influences the choice of yard handling system and vice versa (e.g. Dobner et al., 2000a, and 2003a). On the other hand, we also found that terminal operators – except for terminal operators that control the maritime and continental transportation – have little influence on either the processes or on the information provision regarding the processes before and after the terminal visit of a container. This means that:

- In many cases, joint optimisation of the entire supply chain and processes at an individual terminal is not possible from a single player’s perspective, i.e. the terminal operator’s perspective. Accomplishing a multi-player effort is far more unlikely because the interest of terminal operators, shipping lines and truckers are incompatible with each other. The reason is not so much of a technical nature: agent-based systems already provide satisfying solutions for problems with multiple stakeholders (see e.g. Hengst-Bruggeling, 1999), but more of a political. Power of players is seldom made explicit, and players are not willing to express all their interests.

- The quality of information provided by other parties – customers and non-customers – is a given fact on which the terminal operator has no direct or
very limited influence. The only thing a terminal operator can do is try to stimulate the quality by means of financial incentives.

- The discipline to comply with the planned arrival times at the terminal will always be dependent on external factors, whereas the requirements for service times will remain independent of the realised arrival time. Again, the terminal operator can try to improve the discipline by means of financial incentives.

Thirdly, we observe, at least in Europe, many attempts to improve the hinterland transportation by means of rail and inland shipping (Rakt, 2002). Although the relative share of intermodal inland transportation within the Netherlands remains low, over longer distances the share is significant. In a number of projects (Dobner et al, 2001b), we have observed a trend to anticipate in the planning on a larger share of continental transport via barge and rail in Rotterdam, the latter being fostered by the new Betuweroute to be operational in 2006. In general, this may lead to an increased focus on the rail and barge handling facilities at marine container terminals.

Along with the growth of intermodal in-land shipping, we can observe the rise of so-called integrators, enterprises that focus on the control of more complicated, compound transportation processes (see also Hengst-Bruggeling, 1999). Usually, they rely on a better information management structure than traditional transportation companies, which enables them to provide service that is more reliable and to improve the utilisation of their resources (e.g., Plarina, 2000). Those integrators may contribute to the improvement of the information exchange within the supply chain, creating opportunities to plan landside operations more efficiently.

2.6.2 Three types of design processes

In general, there are three types of design processes regarding a container terminal. The first (Figure 2-13) is the complete new development of a piece of land into a new container terminal, which has no or little interaction with present stevedore activities. The second type of design (Figure 2-14) also concerns the development of a new area, but the operation will interact with current operations.

The third type (Figure 2-15) of design concerns the replacement of equipment at a terminal in operation, altering the control system or changing the layout. Ullmann (1992) addresses these three different types of design issues as design classes, of which the complete new design is the most comprehensive class. However, the
question is whether this is also the most difficult class. In the event that extensions can be carried out without changing the basic processes – both control and execution –, which is mostly the case in conventional terminals, the extension is mainly a matter of putting in more equipment. In an automated environment, it is the same when the control software allows for up scaling. If that is not the case, the extension might be a complicated task. An example in practice is the extension of the ECT Delta terminal in Rotterdam: because the software was not able to control more than eight quay cranes, the extension was implemented as a whole new terminal with separate control software.

Figure 2-13: New terminal development

Figure 2-14: Extension with interaction between existing parts and new parts of the terminal.

Figure 2-15: Redesign of current terminal (e.g., robotization, new control system).

Changing the current system is probably as complicated as building an entirely new terminal or even more complicated. The most important reason is that a change has
to be made in a system operating 24 hours a day, 7 days a week. Therefore, the change has to follow a so-called big bang scenario, which requires extensive testing beforehand. This complication is absent in the event of a new developed terminal: the testing can be done with the real system before the first operation starts. Changes in terms of the physical layout or equipment are hardly feasible in an existing terminal without shutting down the operation. Radical changes in equipment and terminal configuration are rare if not nonexistent in a running terminal, because in most cases it is impossible to implement without shutting down the entire operation. However, ideas are being developed to accomplish this change, forced by the need to increase throughput or service (see Cederqvist and Saanen, 2004).

When we analyse the three types of change – new development, extension, and change of current processes –, we observe a number of differences between the design processes, which are summarised below.

− The scope of the design process. The development of a completely new terminal requires by far the most effort, because it starts with a green field situation and has the most compelling set of feasible solutions. This means that research regarding soil preparation, quay construction, environmental regulations, and so on, has to be carried out. Terminal extensions as well as operational changes only have a small set of feasible solutions because the processes have to be adjusted to each other and the on-going process cannot be stopped temporarily and should be hindered as little as possible. Furthermore, many external circumstances are known and given, which provide the designers with more certainty but also with less degrees of freedom. The fact that one can start with a green field situation also eases the design process, because no limiting factors from or errors in today’s process have to be dealt with.

− The degrees of freedom in the design process. Because extensions and changes have to deal with an existing situation, the degrees of freedom during the design process are far more limited. Furthermore, the current situation forces the extensions to interface with the existing parts, which complicates the design process to a high degree. Especially in design processes, the current situation works as a yardstick; the new situation has to provide a significant improvement compared to the current situation. In new terminal
developments, this yardstick merely consists of reference figures from the performance of other terminals.

- The project risk. Whereas a new terminal development is a project with a high business risk, extensions and changes are of a much lower risk profile. As long as the terminal keeps working during the implementation of changes, the risk is limited, because one can reverse the changes. Most terminal operators as well as shipping lines are risk averse (Saanen, 2001b), especially when concerning automation.

When we analyse the current developments in North-Western Europe – the area where robotization is most likely to be applied in the next 10 years – we see that all three types of design are taking place (see Table 1-4 on page 20), with emphasis on new terminals close to existing terminals.

2.6.3 Design process steps

Rijsenbrij (1999; see also Dobner et al., 2002b) mentions the steps in a container terminal design process up to the start of the technical design (see Figure 2-16). Before paying attention to the factors that influence the terminal facility plan, we make some remarks to the context of this process.

First, it contains all tasks and elements concerning the physical design, including the terminal buildings. However, it does not include the design of the terminal control system of the terminal. Of course, one could argue that this is part of the logistic concept, but in our opinion the terminal control system is as important as the equipment of which the handling system consists. Especially because we have found during our preliminary cases that the interaction between hard- and software is one of the most important design challenges in the design process of terminals that have to meet the increasing requirements, regardless whether the new terminals are automated or not.

A second remark concerns the place of these activities within the entire realisation process of a terminal, which contains more than shown here. Because there will be a need for interaction between the design and implementation teams – which are in most cases different teams, with little overlap –, it is necessary to see the entire process as a whole in order to determine the mutual dependencies, such as
information and knowledge exchange, redesign due to impracticalities and the support when transferring a functional specification into an implementation.

When we analyse the engineering design literature (e.g., Pahl and Beitz, 1988; Tate and Nordlund, 1996; Cross, 1994) we observe a missing link between the elaborately described design activities and the phases after the design is ready, mostly addressed as the operational phase. While it may be the case in an industrial environment that the product has been unambiguously defined after the design activities, it is not the case in container terminal design.
A third remark concerns the market analysis and forecast because the terminal design process takes a long period of time from the first initiative to the commissioning. During this period, the market conditions might change due to a changing economic climate. In order to cope with these external changes, the design process should be unaffected by these types of changes.

2.6.4 Aspects of design-engineering approaches

In literature (Pahl and Beitz, 1999; Tate and Nordlund, 1996; Hall, 1989), we found that most approaches for designing material handling systems consist of multiple activities (in literature often called phases, see Hall, 1989; Pahl and Beitz, 1999; Tate and Nordlund, 1996). In general, four main activities can be discerned:

- Functional design.
- Technical design.
- Implementation and realisation.
- Commissioning and operations.

In literature, the joint four activities are addressed as design-engineering process (Tate and Nordlund, 1996). The important notion is engineering, since a terminal-realisation project contains more than design activities. The whole implementation, realisation and commissioning (together addressed as engineering activities) are indispensable parts of it. Tate and Nordlund (1996) mention a number of aspects, which have to be addressed in a design-engineering approach:

- Decision-making. The purpose of the design process is to make decisions, specifically to find a solution (in terms of a design object) to a certain design problem. Thus, clearly defined decision points and decision criteria (and/or rules) must be visible in the process.
- Performance measures. Performance of the design process is evaluated against the quantity of resources (time, funds) used to satisfy the objective (i.e., solve the design problem). An activity is evaluated against the resources expended to produce its output completely.
- Iteration. The design process includes iteration. That is, similar activities are performed at different points (historical times) in the design process. (This

6 Different names and classifications are used for these activities or phases as for instance Hall (1989) mentions them.
iteration is distinct from that arising due to re-solving the same design problem multiple times.)

− Sequence of activities. Although the individual activities performed are similar throughout the design process, the sequence may differ.

− Levels of scope and levels of abstraction. The design process deals with problems at multiple levels: levels of scope (a measure of the amount of impact the problem has on the overall design) and levels of abstraction (a measure of how conceptual or how detailed the problem is).

− Information management. Data about the design object is collected, generated, used to make decisions, and stored. The information gathered varies in certainty, quantity, and relevance for current and future use.

− Roles and disciplines of people involved. A process is made by people, fulfilling a certain role. In many processes the same role-distribution is applied, which may determine the process execution.

− Role of support tools. In many design tasks tools are used in order to ease and speed up the task, communicate the results or concepts and to store all information created and gathered.

We will see these points back in our proposed design approach, as described in chapters 3 to 6.

2.7 Lessons learnt: The design process of the ECT-DSL terminal.

In the period 1983 until 1993, the design and development process of the Delta Sealand (DSL) terminal of ECT has taken place. In order to get a clear picture of this complex design process a number of key people involved in the design and development process have been interviewed (see appendix 10.2). With all people the same issues have been discussed in order to get a homogeneous, yet rich picture of the situation. The following issues have been discussed:

− Project organisation and role of person interviewed within the organisation.

− Project activities that have been carried out.

− The use of simulation in the project.

− Their experiences regarding the design process.

In this section, a summary of the comments given by the people interviewed is described.
2.7.1 Project organisation and activities

Within the project organisation following engineering groups were installed:

- TOR (Technology and Operations Research)
- PAS (Implementation of Process Administration System)
- PCS (Implementation of Process Control System)
- ASC (Design and realisation of Automatic Stacking Crane)
- AGV (Design and realisation of Automatic Guided Vehicle)
- QC (Design and realisation of Quay Crane)
- Facilities (Design of quay, terminal surface, landside facilities)
- Labour organisation.

The TOR group was responsible for the functional design and for the support of all other design groups. The development of PAS, the administrative part of the terminal control system (TCS), and the PCS, the real-time control part of the TCS, was performed by separate groups. The interface had been specified early in the process, although not all processes had been identified.

The design of the automated equipment, AGVs and ASCs, had been done with limited interaction with the group designing the control software for the equipment. Optimisation between hard- and software design has hardly been possible because of the lack of interaction between the various design groups.

Figure 2-17: picture taken at ECT Delta Dedicated North (DDN, courtesy of ECT).
The project was divided into the following phases:

<table>
<thead>
<tr>
<th>Period</th>
<th>Phase description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1983-1989</td>
<td>Pre-design, handling system concept comparison</td>
</tr>
<tr>
<td>1989-1991</td>
<td>Functional design, technical design of equipment</td>
</tr>
<tr>
<td>1991-1993</td>
<td>Implementation Information System (PAS/PCS)</td>
</tr>
<tr>
<td>1993-1995</td>
<td>Commissioning, fine tuning and improvement</td>
</tr>
</tbody>
</table>

Although the functional and technical design of the control system (PCS and PAS) had to be ready in the middle of 1991, it continued during the implementation, because the functional description could not be fully implemented as written down and also contained so-called blank spots.

The fact that after the commissioning still a lot of design and improvement work had to be done was due to a number of reasons:

- In the design of the control algorithms, especially the stack algorithms and job assignment, the influence of operational conditions, such as stack filling rate, berthing scenarios, varying quay crane productivities, varying vessel arrival times, had not enough been taken into account.

- The measures in the event of equipment breakdown, communication disturbances, and all other deviations from undisturbed operations, were insufficient and often required human intervention.

- The effect of the interaction between operators and the (highly) automated control system had been underestimated, in such a way that all kind of human errors had negative influences on the operational performance.

2.7.2 Use of simulation models

Simulation models were used to support the functional design activities and were partly used for the technical design activities. The TOR group was the only group that applied simulations. At that time, it was not possible to develop detailed models of the entire system, because of the limited computing power available.

Simulation models at different aggregation levels were used. First, a model at terminal level was applied to determine the need for quay length, waterside handling capacity, and stack capacity. The aggregation level of this model is comparable with the model described in section 0. Secondly, a model was applied to compare handling system concepts at an aggregate level. Many present container terminal simulation models (see Liu et al, 2001; Gambardella et al., 1998; Yun and Choi, 1999) have a
similar aggregation level. These models do not contain much control logic; they apply simple control rules and do not represent detailed equipment behaviour. In the DSL project, these models were used to compare a number of different handling systems. Thirdly during the functional design (1989-1991) detailed models of subsystems (AGV system, ASC system) were used to design algorithms like the AGV assignment algorithm, the ASC planning algorithm, the AGV collision avoidance algorithm, and so forth. All of these models were developed in a simulation package called MUST (MUST, 1994), based on Borland Pascal. The programming code of the real planning and control system (PCS) was Borland Pascal as well.

2.7.3 Experiences from the design process

The shared experience of the design and implementation groups was the poor transfer of information from the functional design to the realisation activities. Although the source code of the simulation and the real system could have theoretically been exchanged because they were built using the same programming language, it was not done. Furthermore, interface documents did not contain detailed code descriptions, pseudo code, or a similar description of the system. It contained a verbal description of how the algorithms should work. This led to misunderstandings and deviations from the specifications in the implementation. Furthermore, it led to the observation of the implementation group that many functionally specified components, such as the job assignment algorithm and the routing algorithm, were impossible to implement. Thus, the implementation group redesigned these algorithms, or designed new algorithms and implemented them without functional testing or consulting the TOR group.

After commissioning, and after the fine tuning during the first year of operation, the performance of the system was lower than specified. The following reasons are mentioned for the lower performance:

- Underestimation of the occurrence rate of system failures, such as equipment failures, communication failures, computer hardware failures, and software failures. This was the main reason that insufficient measures were installed in the software to deal with disturbances. Furthermore, failures were seen as exceptions rather than common practice and therefore handled as such. This also explained why the recovery procedures were inefficient.
− Underestimation of the interaction between manual and automated processes. Especially the variation in the equipment behaviour of the manually operated equipment caused difficulties for the automated system.

− The enormous time pressure in the project. During the implementation and realisation phase, there had not been any time to test ready system parts on their contribution to performance. The goal of all activities was to get the system running.

− The fact that the agreed interfaces were a result of a negotiation process between the PAS and PCS group, instead of a rational architecture design meant that they contained compromises.

− The fact that Sealand used the terminal in a different way than originally agreed. The volume they transhipped was much higher than planned, which meant that instead of one-high stacking, the stack height had to be increased to two-high.

− The lack of experience with the design of automated terminals and the tendency to automate the current container business practice, instead of reconsidering the entire concept.

− The fact that the terminal was not designed from a holistic point of view, which meant that the different components had not been sufficiently adjusted to each other.

Based on the experience of the design process of the Delta Sealand terminal, the following points for improvement in the design process were mentioned. These points are also based on an evaluation ECT performed after the project was finished:

− More interaction between people from operations and the system designers, so that during the operation more understanding of the system exists with the operational people and vice versa that the knowledge of the operation is captured in the system.

− A holistic design perspective, so that the system components are adjusted to each other.

− More interaction between the functional design team and the implementation group. In addition, a functional specification containing a format that does not allow for multiple interpretations and implementations and that is understandable to the software development group.
Attention to training of people in the operation to handle and control the automated system. In general, a different type of skill is required for operational people in an automated terminal than in a conventional terminal.

### 2.8 Observations from recent cases

In recent cases we were involved in, we observed various problems concerning the design process of terminal extensions (Dobner et al., 2000; 2001a), new terminal development (Dobner et al., 2001b, 2001c, 2001d, 2001e, 2002a, 2002b, 2003a, 2003b) and the change of terminals (Saanen and de Waal, 2001; Saanen et al., 2002; Saanen and Verbraeck, 2002). These problems are discussed in the next sections.

#### 2.8.1 Gap between functional design and consecutive design steps

A large gap exists between functional design of automated terminals and the technical design and software realisation, in terms of a lack of information transfer, hardly any involvement of the staff from the functional design team during the technical design and software realisation, and a lack of functional requirements that can be transferred into software, without ambiguous interpretation. This problem is less common in equipment design because there is a much longer tradition of equipment design, and the used components are already in a mature stage of their lifecycle (see also section 2.7).

#### 2.8.2 Non-integrated design of hardware and software

There is insufficient interaction between the hardware design of robotized equipment and its control software, leading to sub-optimisation of each component. Even the equipment design is fragmented: electrical design, mechanical design, and structural design: all lead to different solutions for single problems.

#### 2.8.3 Insufficient attention for interaction between operator and automated system

In the design process, little attention is paid to the interaction between the operator of the automated system and the system he controls, i.e. the equipment. This may lead to a user interface that contains functionalities that cannot be used properly by the operator, because he does not get the right feedback, or he is not able to understand the consequences of changing a particular setting. As a result, he is likely to operate the system in a suboptimal way as we found as well.
2.8.4 **Gap between strategic and operational performance indicators**

A gap exists between aggregate, strategic targets, like throughput volumes and vessel service times, and operational, day-to-day, hour-to-hour operational targets, such as quay crane productivity and truck service times. Moreover, there is no standardised way of translating these strategic goals into workable operational goals (Dobner et al., 2002b).

2.8.5 **Lack of tools to analyse operations at robotized container terminals**

There are no tools available to provide insight into the operation of automated equipment and/or automated terminals, including solutions for process control systems. The tools either are of a static nature or can only deal with manually operated equipment. The simulation tools that are available to analyse container terminals mostly do not contain the specific logistic control rules required in the real operation and are therefore a rather optimistic representation of reality.

2.8.6 **Unavailability of COTS solutions for robotized terminals**

There does not exist a common-off-the-shelf terminal operating system (TOS) for automated terminals (see also section 2.4), which increases the risk of realising an automated terminal. The only two terminals that have automated transportation and yard equipment – ECT and CTA – developed their process control systems themselves. The large TOS providers only offer packages for manually operated terminals yet, at least with regard to horizontal transportation and yard handling.

2.8.7 **Lacking integration between performance and cost analysis**

There is a lack of integration between cost analysis tools and performance analysis tools (Dobner et al., 2002b). Usually the trade-off between cost and performance is non-linear, which complicates the design of cost-effective solutions without the use of tools that can cope with non-linear behaviour. Furthermore, in order to create a realistic operational picture, on which a cost analysis can be based, the model of the operation has to be comprehensive.

The following example (based on Dobner et al., 2002b) illustrates the non-linear relationship between cost and performance. The data result from a comprehensive analysis of various terminal concepts, supported by an analysis by means of simulation (see also section 4.2.7) and an extensive cost model. We combined the
cost data and performance data in Figure 2-18, which show the total capital cost of the equipment, calculated per operating hour, and shown for AGVs and RMGs. We only took the capital cost, because when increasing the amount of equipment with the yearly volume staying at the same level, the total number of operating hours stays approximately at the same level because only in a peak situation these additional pieces of equipment will be in operation. However, the quay crane productivity increases when the supplying system is extended – the number of AGVs increases from three per quay crane to eight per quay crane – as can be seen from the productivity curve: an increase from 69% (of the maximum achievable productivity) to 98%. However, the marginal benefit decreases. The cost of equipment increases linearly, so we assume that the total operation cost of the quay crane decrease when the productivity increases, because the number of operating hours decreases. The possible gain due to the vessels being in port for a shorter time has not been taken into account, because these gains do not directly contribute to the terminal operator’s profit. In Figure 2-18, we see that upward from six AGVs per quay crane the total cost increases, although the performance increases as well, however the benefit from the performance increase is lower than the increase of capital cost.

Figure 2-18: Example of the relation between cost and performance; the total capital cost of the operation per quay crane are shown here in a peak situation.
2.8.8 Design-engineering process ends after go-live

Current design approaches do not address the activities after commissioning, apart from monitoring and post-evaluation. Current practice is even worse; after commissioning terminals struggle with operational problems, but do not evaluate or look back. Without a clearly defined relationship between goals and measured performance analysis afterwards, it is hardly possible to improve a terminal’s operation in a systematic way.

Based on our findings from the inductive cases, and a literature review, we have formulated a research approach, as will be described in the next section.

2.9 Research approach, boundaries, and instruments

2.9.1 A simulation approach as starting point

Based on our inductive analysis, as described in the previous sections, we have formulated our research objectives. The main objective of the research is to develop an approach for designing robotized, marine container terminals, which addresses the specific characteristics of such terminals, and considers the specific properties of terminal environments. Given the problems and pitfalls observed in the inductive cases (see sections 2.7 to 2.8), and literature on this subject, we consider it a valid research objective to develop an approach to design especially robotized terminals in a more systematic way, addressing and solving problems that we encountered during the inductive cases.

According to Holbaeck-Hanssen et al. (1975) it is necessary to have suitable tools for system description in order to be able to understand, design implement or control complex systems. By writing a system description, the inquirer forces himself to consider relevant aspects of a system, and a system description language should be so constructed that it assists him in this process. By writing a description, the inquirer makes it possible to convey his conception of the system to other people. Thereby he may contribute to their knowledge, and make it possible for them to correct his views and to improve his understanding (cited by Sol, 1982, p. 19).

As starting point of our approach, we propose to use a simulation approach (or problem-solving approach, or model based approach, Rosenblit, 2003). We take this approach as starting point, because it provides an extensively tested approach (e.g. Sol, 1982; Meel, 1994; Meinsma, 1997; Janssen, 2001), in which:
The dynamics of a system are explicitly taken into account, which is important for a design approach of a system in which the dynamics and uncertainties largely influence the behaviour and performance.

Comparison between different solutions (alternatives) is an explicit step. Because in terminal design there are many possibilities with differing properties without a clear “best” alternative, comparison between alternative solutions is one of the major steps in the design process.

A support environment (a model or model suite) is one of the main tools to express the properties and behaviour of a (future) system.

Although a problem-solving approach is common during the design of a system, it is less common during software implementation, testing and commissioning. In software engineering, two widely used approaches are the sequential and cyclical methodologies, informally known as the waterfall (e.g. Boehm, 1979) and spiral (Boehm, 1985) methodologies (Burbach, 1999, p.4). They are generic in design and have been simplified to emphasize a key aspect. In a sequential methodology, the four phase of analysis, design, implementation, and testing follow each other sequentially. In a cyclical methodology, the four phase of analysis, design, implementation, and testing are cycled with each cycle generating an incremental contribution to the final system. There are various other methodologies (see Burbach, 1999), which we will not discuss here in detail.

The purposes of the use of models are the following (Booch, Jacobson and Rumbaugh, 1998:13):

- To capture and precisely state requirements and domain knowledge so that all stakeholders may understand and agree them. The interesting thing here is the assumption that all stakeholders could possibly understand or even agree on the models. In practice this is not always possible, because modelling contains always an element of subjectivity, albeit the modeller’s way of representing and depicting a real thing.

- To think about the design of the system. In principal a model is a simplified representation of reality or future-reality and that enables an analyst, a designer or a constructor to investigate the subject in a cost-efficient way. Here for instance, performance optimisation is one of the key objectives.
− *To generate usable work products.* Because models can be made in an early stadium, they function as source of inspiration for new alternatives as well as way of analysing possible consequences of the choice that have been made or are being made.

− *To organize, find, retrieve, examine, and edit information about large systems.* As systems get more and more automated as well as more complex due to their scale, intelligence and so on, the need for tools, i.e. models, rises, that enable the stakeholders to define views or aspects of the system and model them individually as well as in coherence.

− *To explore multiple solutions economically.* Especially in case there are hardly any similar systems in existence that can serve as calibration – i.e. a best practice –, the need for models as way of analysing and evaluating alternatives arises. With those models, we are able to provide insight in the consequences of possible alternatives before implementing them in reality.

In order to be convincing, this insight the models provide has to be of a high quality. Especially in this area, where the processes are of a complex nature – dynamic, uncertain, mutually dependent –, where conventional systems are about to be replaced with new, automated systems, and where the decision-makers tend to be very risk-averse, it is a challenge to obtain the required level of quality.

Quality aspects are reliability of the insight – mostly expressed in estimations of the performance and behaviour to be expected –, the validity of the insight based on the results from the analyses and the credibility of the insight. The situation is even more complicated when completely new concepts are introduced, because validation then cannot take place by comparing the results with current practice. Therefore, the following additional research objectives and questions can be defined:

− How can we enclose the (new) specific properties of processes at a maritime automated container terminal into the support environment that we use to gain insight from and to perform the analyses?

− How can we reduce the risks of robotization and automation by means of a design approach?

− How can we ensure that the insight we provide is reliable and valid? How can we validate results when we do not have similar examples that are already operational?
– How can we ensure that the design approach is applicable to the portrayal and analysis of new (innovative) processes, enabling new and old processes to be compared in the same formats and perhaps even driven by the same set of variables?

2.9.2 Research boundaries

The research will focus on the design of marine container terminals. This does not mean that the findings from this research are not applicable in other domains, especially in the field of transportation and logistics. On the contrary, we already have observed many similarities in other fields – for instance the design-engineering of airports (see Rengelink and Saanen, 2003).

Figure 2-19: One of the largest maritime container terminals worldwide: Container Terminal Burchardkai in Hamburg (left picture), and the latest robotized terminal: Container Terminal Altenwerder, also in Hamburg (right picture).

In the preceding sections, the main properties as well as the context of the design process of marine container terminals have been described. We found that principally three types of design processes should be supported, i.e. new terminal development, the extension of existing terminals, and the change of existing terminals. Furthermore, we discussed the wide variety of terminal configurations (section 2.2), which can partly be explained by the location and the volume flows through the terminal. However, our approach should not be affected by a terminal’s size, location or other site-specific properties. Finally, we put emphasis on the trend of robotization. Since most terminals are still manually operated, and the near future will bring more robotized concepts, the design approach should be applicable for both types of operations.
The research objective aims at the development of an approach for designing (automated) marine container terminals, in order to answer on the one hand the increasing demands from the operators, and on the other hand an effective and efficient integration of new technology in terminal operations.

A major element of the dynamic modelling approach that has been used as basis for the design approach is a support environment used to create insight in both the IST, i.e. the current, and SOLL, i.e. the desired, situation. The support environment consists of building block oriented simulation models (Verbraeck and Dahanayake, 2002). This means that other types of tools have not been taken into consideration. The choice for applying computerised models or simulations is discussed below.

First, the application domain concerns a type of operation that consists largely of standardised and repetitive tasks: transhipment, transport, and storage. Secondly, these tasks require some skills but limited decision-making intelligence from the operators. Thirdly, the different tasks that together contribute to the overall system performance are dependent of each other and, therefore, require integrated planning in order to accomplish an efficient system. Fourthly, the planning is a task that requires overview of the terminal operations, understanding of the relations and interactions, and eventually optimisation techniques. Finally, due to all kind of external influences – weather, human skills, congestion, insufficient information quality – the tasks all have a stochastic character, in the sense that the exact duration, sequence and location are difficult to determine. Therefore, this uncertainty must be taken into account. Dynamic simulations are suited for modelling of processes or tasks that meet the characteristics mentioned above (Meinert et al., 1998). Furthermore, they explicitly consider uncertainty by means of stochastic distributions. Other reasons for applying simulation are mentioned by Wilcox et al. (2000):

- It is a cost reducing way to develop new systems and to train people to operate systems, especially because it is possible to generate all kinds of exceptional situations in an accelerated mode.
- It overcomes a number of impracticalities involved with testing in real-life situation. Possible impracticalities are the lacking of physical equipment and non-occurrence of disturbance situations.
The choice for building-block-based simulation models can easily be explained. Since one of the main objectives of this research is to speed up the process of gaining insight and supporting the (re-) design process, the application of component-based or building block-based models has already proved its potential and added value in earlier projects concerning design of complex systems (Saanen et al., 2002c; see also Davis et al., 2000). The essence of building-block oriented models is that the building blocks are self-contained and defined by their interface through which they provide a particular service (Verbraeck et al., 2000). As long as the interface remains unchanged, the behaviour within the component may change, without affecting other building blocks. We also see that in adjacent areas, such as distributed simulation (Wilcox et al., 2000) the target of re-use, realised by component based models or software, is set and successfully met.

2.9.3 Research instruments

Research instruments are used to describe the way the data is collected and analysed (Janssen, 2001). Various research instruments are available, such as experimenting, correlation study, naturalistic observation, surveys, and case study. In this research the most important research method is the applied case study, also addressed as action research (Janssen, 2001, Checkland, 1981). Applied case studies, in comparison to case studies are more focussed on the “how to?” question (Meel, 1994), instead of the “why?” and “how?” questions. Therefore, instead of only explaining and analysing, we emphasize the design of a design approach. Therefore we are addressing the “how to” question rather than the “why?” and “how?” questions. Of course, in order to answer the “how to” question the other two questions have to be addressed as well.

Yin (1989) defines the case study as an empirical inquiry that investigates a contemporary phenomenon within its real-life context when the boundaries between phenomena within its real-life context are used and when multiple sources of evidence are used. Moreover, he describes the variations in case studies as a research strategy. Yin also notes that case studies are the preferred strategy for studies dealing with “how” or “why” questions. Case studies are also used when the investigator has no control over the events. According to Sommer and Sommer (1991), case studies provide greater depth to the investigation. They tend to maintain the integrity of the whole with its myriad of interrelationships. Case studies also give the investigator the
opportunity to apply a multi-method approach to a topic. They are not without their limitations, because many case studies take place after the fact. They depend on people’s memories, which can be faulty. Case studies are also difficult to repeat and the possibility to generalise is limited (Sommer and Sommer, 1991; Yin, 1989).

Yin mentions the following guidelines for case study research. Firstly the use of multiple sources of evidence, secondly the recording of data and observations in a case study database, and thirdly to determine the chain of evidence. The use of multiple sources of evidence enables the development of converging lines of inquiry, a process of triangulation (Yin, 1989, p. 97) and addresses the problem of construct validity by providing multiple measures of the same phenomenon. The second guideline aims at organising and integrating data. Fielding and Fielding (1986) point out that the use of multiple sets of data is not enough to avoid the problems of bias. The problem, according to them, “is not to use several approaches alongside one another, but to achieve integration” (p. 26). The integration of various data increases the reliability of the study. Finally, Yin mentions the necessity for a chain of reasoning by cross-referencing all aspects of the research process (e.g., research questions, methodological procedures, and resulting evidence). The goal is to establish “explicit links between the questions asked, the data collected, and the conclusions drawn” (p. 84). Thus, a reader can “follow the derivation of any evidence from initial research questions to ultimate case study conclusions” (p. 102). As Yin has pointed out, the chain of evidence can help to enhance both the reliability and the construct validity of the case study. According to Christie et al. (2001) a case study methodology should be chosen when:

− There are particular events that are focused on a situation or context and have specificity (Eisenhardt 1989; Merriam 1988). In each different container terminal design project this is the case; circumstantial settings are different (Hekman, 2001), requirements are different, et cetera.

− The social organisational settings are complicated (Kaplan 1986; Morgan and Smircich 1980; Orlikowski and Baroudi 1991; Parkhe 1993). This is also the case in maritime terminal design projects, since (1) multiple disciplines are involved and (2) the process takes a long time.

− The researcher seeks contextual meaning within a bounded system (Bonoma 1985; Stake 1978; Yin 1989). Although the research is focused on the container
terminal as bounded system, its interaction with the entire supply chain has to be taken into account, since the terminal plays an important role as link between the intercontinental and continental transportation processes.

– The research enterprise is inductive theory building (Gilmore & Carson 1996; Hirschman 1986; Merriam 1988). Given the research approach, this is the case since we depart from findings and observations and try to formulate a thesis on that, which then is applied and tested in practice.

Given the conditions above, the type of research here is very well suitable for case study research. Therefore, the case studies performed will form an important pillar in both the exploration and testing phase of the research. In the first phase, they will provide a rich picture of the current container terminal design process as well as an overview of the terminal designs themselves. In the second phase, they will provide feedback and an empirical environment for testing the proposed design approach.

The *model systems* that we have developed as an integral part of our design approach could also be considered as *research instruments*, as they also serve to analyse processes at container terminals and serve as decision-support system during the design process. They have played an important role in all case studies – both the inductive cases and the test cases – and reflect our view on container terminal processes in terms of the way the models are structured and in the way the models are a reduced representation of reality.
3 Essentials of an Approach for Designing Robotized Container Terminals

In the two preceding chapters, the context of the research has been described. Moreover, the inductive cases have created a picture of the main problems that have to be addressed when designing and realizing a container terminal and, in particular, a robotized container terminal. In this chapter, the specific properties of a container terminal are more closely discussed. Furthermore, we describe the requirements to the design approach as resulting from the inductive analysis in the previous chapters. Then, the premises on which the design approach is based are elucidated. This chapter ends with a set of guidelines that form the framework of the design approach.

3.1 Characteristics of a container terminal

A marine container terminal can be considered a complex system. Although many discuss complexity (see Hall, 1989), the definition of complexity is mostly of a descriptive nature; “unprogrammable decisions” (Simon, 1960), “wicked problems” (Rittel and Webber, 1972) and “double-loop problems” (Argyris, 1982; Argyris et al., 1985). In order to clarify what we mean with complexity in the context of the design of marine container terminals, we will discuss various aspects of a terminal’s complexity in the following sections. Because these aspects have to be addressed in the design approach, we will return to them in the next chapters, where the design approach is described in more detail.

3.1.1 Conflicting requirements

The complexity of a system increases when the design requirements (or objectives, see Hall, 1989) (partially) conflict, in the sense that they are hard to align. This often leads to a final design containing compromises. Furthermore, in order to accomplish a compromise, trade-offs have to be made between unequal quantities, such as environmental pollution and productivity. These trade-offs can be made by using multi-criteria analysis (MCA), although the units by which objectives can be quantified are incomparable. Therefore, any MCA will contain the subjective

---

In our definition of a system we follow Hall (1962) in his definition: “A system is a set of objects with relationships between objects and between their attributes.”
transformation to a comparable unit (currency or utility), which might lead to different results when applied by people that act from various standpoints.

In the case of container terminals, there are certainly conflicting design requirements: handling performance, cost reduction and minimisation of land-use are already hard to align, not to mention environmental aspects, such as noise and air pollution.

3.1.2 Influence of external events and circumstances

The complexity of a system increases when external events and conditions increasingly influence the functioning of the system. The arrival of vessels, trucks or trains can all be considered as external events because these events are beyond the span of control of the container terminal operator. The terminal operator has little control over these external events and information about these events is usually insufficient or of poor quality. The only activity a terminal can perform internally is housekeeping to get prepared for the next operation.

The effect of changes in external events to the terminal’s processes is great. When one vessel arrives late and at the same time another vessel is present at the quay, the late vessel has to be served because of the service contracts with the shipping lines. This causes peaks for the yard handling system and the transportation system. Hence, the decoupling of external and internal processes is lacking: at a marine container terminal, the external processes dictate the internal processes.

3.1.3 Information about external processes is poor

Besides the control of external events being beyond the control of the terminal, the information the terminal receives about the external events and circumstances – such as peak loads – is often lacking or unreliable. This is, for instance, the case with vessel arrivals, container information, truck, train, and barge arrivals. Part of the productivity loss is caused by external changes: a package of containers has to go unexpectedly with a vessel, this package is not yet at the terminal but the vessel is already being loaded, or the container weight and destination are incorrect so rehandles have to performed in the stack; and so forth. In general, the information a terminal possesses in advance is poor in terms of both quality and quantity (Saanen, 2001b, 2002a). This increases the complexity to control and optimize the operation.
3.1.4 Multiple interaction points between system and external world

A container terminal is the meeting point of multiple flows, and the connection point of multiple modes of transport. Because all modes have to be handled at a container terminal (i.e., deep-sea, short-sea, feeders, barges, trains and trucks), the possible occurrence of simultaneous events (see previous section) is a day-to-day event. Each flow is handled at a different part of the terminal (berth, gate, rail terminal). When the peak of these processes coincides, it can result in peaks at different locations of the terminal. This can create unexpected side effects; because all processes meet at the yard and at a certain point, the yard handling equipment can no longer handle the demand. In such cases, this can influence all other processes at the terminal, like the vessel service, the gate, or the rail terminal.

Average productivity = 40 cycles per hour

![Figure 3-1: Typical quay crane cycle time distribution (Saanen and De Waal, 2001).](image)

3.1.5 Processes with large variations in duration

The complexity of a system increases when the processes increasingly show varying outcomes. In most terminal processes, the variation of possible outcomes is large. This means that during an operation the individual outcomes of a number of processes vary heavily. Take for instance the cycle time of a vessel discharge or loading movement; in most cases, this cycle time varies between 40 seconds and 5 minutes and it is impossible to predict the duration in advance (see Figure 3-1), better than the average value. Another example is the arrival of vessels; they can be
eight hours early but also four days late, while sailing on a weekly schedule. Most processes at a container terminal have a large variation in terms of the deviation from the average value of the process time. Moreover, at a micro-level, the variation in the processes influences the entire operation, especially when they are automated. In the case of manual processes, the drivers solve some micro-level adjustments themselves by allowing overtaking, last minute changes, or rerouting in case of incorrect information, such as wrong stowage, or crane movement.

The large variation of the process limits the possibility to plan an operation in advance, because the varying outcome can change the execution from what originally had been planned, based on the best possible prediction, which is in most cases the average. Robust planning approaches in which the possible outcomes of any stochastic process are considered in the decision-making, address this issue by various alternatives such as sensitivity analysis, stochastic dynamic programming (Koole, 2002) approach, and min-max regret approach (Kouvelis and Yu, 1997; Averbakh and Berman, 2000). However, in robust planning literature (see e.g. Koole, 2002; Wang and Yu, 2002), it is recognised that the size of the problem solving space is often too large to provide practical solutions in real-time.

3.1.6 Interlinked processes

The main processes at a container terminal (i.e., transhipment, transportation, storage, and an additional process such as the customs inspection) are interlinked with each other. A process is interlinked when timing, sequence, duration, resources, and output of one process influence another process and vice versa. At a terminal, there are a number of interlinked processes, such as the vessel arrival, the vessel handling process, the horizontal transportation, and yard handling. The main decoupling at a container terminal is the stack (yard), which is an indispensable buffer between the waterside arrival pattern and the landside arrival pattern, between which dwell times occur varying from as little as a few hours to as long as several months. In case of late arrivals (i.e., a container that arrives when the vessel is being loaded), the landside handling is directly linked to the waterside process because the movement into as well as the movement out of the stack are both within the same planning horizon.
3.1.7 Non-linear interactions between processes

As said above, the processes at container terminals interact with each other, in the sense that the outcome of one process affects the outcome of another process. Moreover, due to the varying outcome of processes, the interaction of two or more (sub-)systems results in losses (i.e., waiting times). The current practice at a container terminal shows that operators solve dynamic effects by using far more equipment than would be necessary from a static productivity point of view (Dobner et al., 2002b). However, to improve efficiency it will either be necessary to increase performance using the same amount of equipment or to reduce the equipment pool to a level closer to what would be required from a single-system productivity point of view. In order to accomplish this, the interactions have to be controlled in such a way that the variation of one process has no (or a minor) effect on the process it is interacting with.

![Quay crane productivity curves for various yard and transportation equipment configurations](image)

Figure 3-2: Quay crane productivity curves for various yard and transportation equipment configurations (Dobner et al., 2001b).

One of the properties of relations between processes is the fact that most relations are of a non-linear nature. Recent studies (Dobner et al.; 2001a, b, c, d and 2002a, b) have clearly shown (1) that the various processes are interdependent and (2) that the relationships in terms of productivity are non-linear. When, for instance, the productivity of a quay crane is analysed in relation to the amount of equipment
serving this crane, we see a non-linear relation between the performance and the amount of equipment used (see Figure 3-2). The productivity represented, results from a combined process of stacking cranes storing and retrieving containers into and out of the stack, and automated vehicles (AGVs) transporting containers from stack to quay crane and vice versa. When we then analyse the ratio between berth productivity and the number of stacking cranes and/or the number of transportation vehicles, the following performance curves can be constructed, which clearly show the non-linearity of the relationship (see Figure 3-3).

![Figure 3-3: Performance graphs for varying numbers of transportation vehicles and stacking cranes (Dobner et al., 2001b).](image)

### 3.1.8 Planning and co-ordination of terminal processes

The complexity of single processes – mainly due to both the external and internal unpredictable influences – is such that the processes have been divided into sub-processes in order to create feasible tasks, feasible in terms of computing time or human span of control. Examples at a container terminal are a vessel discharge process, or a stack reorganisation process (housekeeping). Each of them has been divided in a number of individually controllable tasks, such as quay crane movements, transport movements, and stacking crane movements. The movements are planned, co-ordinated, and controlled. The total integrated planning and optimisation of these processes, is a task with many solutions.
The current terminal control systems show a clear cut in sub-tasks, such as berth planning, vessel stowage, crane planning, yard planning and planning of horizontal transportation, even in case of computer-supported planning processes. Integration of these sub-tasks might lead to a high burden on the control software and yet have a small benefit. The fact that the benefit is low is due to the high unpredictable variations of various processes (see previous sections), making optimisation hard. Stochastic dynamic programming is an approach that can be used to take decision under dynamic situations (see Koole, 2002). However, as the many processes at container terminals are all of a dynamic nature, the combined possible states of each process create an N-dimensional problem, which requires high computation power to be solved. As the decisions have to be taken in real-time, the approach appears to be unpractical for application at a container terminal. Therefore, the planning is done on a basis of splitting of tasks and co-ordination (i.e., mutual adjustment of processes in time and place), rather than optimisation. However, the overview of all processes is then difficult to get, and actions at one point in the terminal might result in unexpected consequences for other processes in the terminal.

3.2 Characteristics of a container terminal design process

In addition to the complexity of the design subject, i.e. the marine container terminal, the design process contains a number of complexities as well. This type of complexity is addressed as process-related complexity. In order to clarify what we mean with process-related complexity, we mention a number of elements that increase the complexity of the design process (see also Figure 3-4):

- The duration of the design process. The process from first idea to commissioning usually takes six to seven years, including the applications for environmental licenses. An average design process of a marine container terminal takes at least three to four years (Rijsenbrij, 2000). During the process all kind of things change; e.g. the global and regional economical forecasts, the available equipment technology, the environmental regulations. These changes have to be incorporated in the design process and are likely to lead to changes in the design during the design process.

- The number of people involved in the process. In chapter 2, the multi-disciplinarity of the design process has already been addressed. Furthermore, the number of people involved easily exceeds 50-80 during technical design and
implementation, excluding the people used form suppliers. The more people involved, the more difficult the process is to manage, especially when multiple disciplines are involved as well.

− The number of stakeholders involved in the process. Usually there are three main stakeholders: the port authority, the terminal operator, and the shipping line. However, within the design process itself, more stakeholders are involved, ranging from civil engineering companies to software companies and from labour unions to environmentalists. The more stakeholders, the more interests differ, which can potentially lead to varying perceptions of bottlenecks and solutions.

− The activities in the design process that are distributed over time require different expertise. Not only are different disciplines required to design a terminal, the expertise per discipline varies per activity. Functional design requires thinking at a different abstraction level than technical design; the same holds for development, implementation, and commissioning.

![Figure 3-4: Characteristics of design activities: the activities differ in terms type of people involved, scope of design and the available information.](image)

− The size of the project and as a consequence the business risk. The investment cost of a large deep-sea container terminal is in the range of €100 Million to €500 Million. With an increase of robotization, the capital cost increases due to the higher equipment cost. The robotization itself increases the business risk as well. Because robotized terminals are still not considered proven technology, the risk is perceived as significantly higher than the risk involved with
conventional terminals. Furthermore, with the increasing scale, the terminals increase in size and amount of equipment as well. Both trends heighten the risk involved.

3.3 Framework for a new approach for designing marine container terminals

3.3.1 Main design activities and their basic properties

The four main activities that were discerned in chapter 2, form the basic structure, because they show principally different properties in the sense that during these activities different questions are answered.

- Functional design.
- Technical design.
- Implementation and realisation.
- Commissioning and operations.

In the functional design, the why- and what-questions are answered without going into the details of the how-to-question, which means that emphasis is put on the problem (why), the objectives (what) and the purpose of the project and the type of system that has to be implemented (see also Hall, 1989). The why and what-question can be addressed at multiple aggregation levels, from the objective and function of the entire system (‘holistic view’) to the objective of each system component like the gate or a piece of equipment. This decomposition can be addressed as hierarchical design, which is according to Hall (1989) one of the main properties of functional design.

The bow-to-question is answered during the technical design, during which a translation is made from functional specifications (‘what’) to a technical solution (‘how-to’). As for the functional design, the technical design also takes place at multiple aggregation levels; from the design of the entire logistical flow through the terminal and the choice of the handling system, to the design of a crane beam or a dispatching algorithm for transportation vehicles.

With aggregation is meant that components of a terminal can be aggregated to systems of multiple components. The level of aggregation determines which subcomponents are considered as one component. The more components are considered as one, the higher the aggregation level. The lower the aggregation level, the more components are considered separately.
The implementation is characterised by development and building, and with technical design at detailed level (also ‘how to, but at implementation level’). This activity still interacts with the functional and technical design because during the implementation still white spaces can appear.

During commissioning and operations, the testing, use and evaluation of the developed system is the main activity. As we will discuss further on in this chapter, the design process should not stop after commissioning, although the intensity and cost may decrease as soon as the system meets the requirements. However, in order to stay in tune with the external circumstances – handling volume, call patterns and landside arrivals –, the terminal’s operations should be monitored and adapted to the changing conditions, whenever required.

**Figure 3-5: Characteristics of the design-engineering approach**

In literature, the four activities are usually described as sequential and partly overlapping phases. However, in practice we observe that the functional design activity lasts beyond the start of the implementation. Secondly, during the functional design it is often necessary to go into detailed design issues to limit the set of feasible alternatives or test the feasibility of the decisions on an aggregated level. Furthermore, during the design-engineering process, the external conditions might
change as well, which might result in adjusting the requirements. Figure 3-5 represents the design-engineering approach to which we adhere.

The following remarks can be made about this design-engineering approach:

− The various design activities address design issues at varying aggregation levels, also associated with the range from strategic questions to operations questions. The interaction between design issues can either be top-down, bottom-up or middle-out, which means that not all questions start at the highest level of aggregation. In many cases, detailed design concerns are the starting point for a design loop that ends with aggregated issues. However, we suggest that early in the process an overall architecture is constructed that fulfils the role of stepping stone. The architecture structures the whole of functionality into logical components that have an interface through which their functionality can be triggered. Preferably, this architecture remains untouched as much as possible throughout the design process. The implementation within a component may evolve during the process (mostly from aggregated to detailed) as long as the interface remains unchanged. Typical design issues at various aggregation levels are mentioned in Table 3-1.

− The process of functional design, technical design, implementation, and commissioning should be executed as an iterative process, so that findings during the technical design, implementation, or even commissioning are fed back to the functional design.

− During the design process, the level of detail increases, because more information gets available and problems that are more detailed have to be solved. However, system performance and functionality – for instance, the type of service offered to customers, the achievable vessel turn-around times – remain on the agenda, because the attraction of volume is an ongoing process.

<table>
<thead>
<tr>
<th>Strategic</th>
<th>Tactical</th>
<th>Operational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service towards customer</td>
<td>Type of handling system</td>
<td>Type of equipment</td>
</tr>
<tr>
<td>Type of terminal (OD of hub-centre)</td>
<td>Lay-out</td>
<td>Equipment specifications</td>
</tr>
<tr>
<td>Manually operated or robotized</td>
<td>Logistics control like planning of berths, cranes and yard</td>
<td>Control algorithms like routing, job assignment.</td>
</tr>
</tbody>
</table>

In our approach, we explicitly avoid the distinction of design phases, although it is quite common in design-engineering literature to distinguish phases rather than
activities (Tate and Nordlund, 1996). The reason to distinct activities, which do not immediately link a timeframe to the various activities in a design process, is that they justify reality better in terms of system and process complexity. Especially the process complexity makes it difficult to divide the process into a number of sequential phases that suggest that for instance all of the functional questions will be answered during the functional design. In practice, as the ECT-DSL case has clearly shown (see section 2.7), many design and engineering activities are executed in parallel, although the project plan foresaw in a sequential, partially overlapping process.

![Diagram](image)

Figure 3-6: Context of approach (see also chapter 6)

The approach we propose should be applicable to both iterative (multiple loops through design activities) and interactive (interaction between various parallel design activities) design processes. This means that it should not prescribe a certain sequence and it should allow for changes during the entire process. However, it is important to determine (see Figure 3-6):

− The scope of each activity within the design process (see chapter 4).
− The way decisions are made, and how performance of alternatives is measured.
− The information that is needed for the design and decision making process.
The way models (e.g. the suite of simulation models) can be applied (see chapter 5).

The role of people involved (i.e. designer, decision maker) in the design and decision making process (see chapter 6).

3.4 Starting points of the design philosophy

3.4.1 Starting points of the design philosophy
Based on our inductive research (see chapters 1 and 2), a number of problems and challenges have been identified in the area of the design of robotized marine container terminals. These problems and challenges form the requirements for the approach we propose. The way the design approach addresses these problems is given in the form of guidelines, which are described in the following sections. The elaboration of these guidelines follows in chapters 4, 5, and 6. The guidelines are applied in various cases, which are discussed in chapters 7, and 8. In this section, we firstly discuss a couple of starting points that are the basis of the design approach.

As a start, we depart from the assumption that improving the design approach, will also lead to a better fulfilment of the requirements set to the terminal design itself. We experienced in various cases (Saanen and Waal, 2002, Dobner et al., 2001a, 2001b, 2002a, 2003a, 2003b) that the focus in container terminal extension projects is mainly on the solutions, i.e., the technical systems. We argue, that besides the solutions discussed, the design process, i.e. the way solutions are conceptualised, specified, evaluated, chosen and implemented, is as important as the solutions themselves. The focus in this thesis is on the design process, not on the specific solutions. Although the objective of the research is not to design a more efficient container terminal, the objective behind is to design terminals that are able to meet the increasing requirements, without a loss of terminal efficiency.

The second starting point is to apply a simulation approach throughout the entire design-engineering process. Sol (1982) and Roekel et al. (1999) address this extended use of simulation as a new paradigm for system development. This approach is favoured over other approaches, because the interaction between the components of a container terminal cannot be captured by analytical models (Piera and Guasch, 2002). Conventional planning tools that handle scheduling problems by using analytical techniques often fail to catch the appropriate level of detail when applied to
the design of a container terminal handling system. For instance, queuing theory methods (see also Hilkens, 2002) can model steady state operations, but they fall short to deal with transients. Hierarchical Planning Tools offer good results to deal with complex problems, which might be decomposed in “independent” subtasks (i.e., short-, medium-, and long-term production planning) (Sethi, 1994). However, these tools are inadequate to solve internal terminal design problems, where the berth assignment, available quay length, and available space for container storage cannot be treated as independent targets (Piera and Guasch, 2002). A recent investigation (Hilkens, 2002) shows that a mathematical model, using queuing theory, requires major simplifications of the system which has the following effects:

- The inability to analyse certain interactions.
- The inability to visualise the operational process.
- Results that are hard (if not impossible) to validate.

Because static analyses do not consider the various interdependent stochastic processes, the use of simulation is seen as by far the best way to capture the integrated design complexity of the combination of terminal geometry, logistical concept, and handling system (see also Piera and Guasch, 2002).

Therefore, we assume (1) that the application of a simulation approach in each activity of the design-engineering process, contributes to the duration of the design activity, to the understanding of the system’s behaviour and to the quality of the system (and its components) in terms of the terminal’s objectives. Moreover, we assume (2) that the application of the same kind of simulation models throughout the design-engineering process will provide synergetic effects in terms of a reduction of development time and an increase of the quality of the models.

Thirdly, we assume that a building-block-based model architecture\(^9\) (Verbraeck and Dahanayake, 2002) contributes to the feasibility of re-using models throughout the process. In such a way, it is possible to use the same components (building blocks) for various purposes and to enable the use of components at various aggregation levels within a whole. Such an approach allows designers to develop certain building

---

\(^9\) A building block is a set of objects that provides a service (Jacobs and Verbraeck, 2002).

\(^{10}\) A system’s architecture is defined as a conceptual model of the structure of a system and the interrelationships among its building blocks. For further explanation see chapter 5 (see also www.sei.cmu.edu/architecture/definitions.html).
blocks, validate them, and use them for multiple purposes, without having to bother about validation again.

3.4.2 Use an object-oriented world-view
Our first guideline is to use the object-oriented modelling paradigm for all modelling activities, which means that the entities that execute actions are leading. The object-oriented modelling paradigm has a number of advantages (Booch, 1986), (Rumbaugh et al, 1991), which makes this way of viewing the world suitable for a terminal design process. When the object-oriented way of modelling is compared with the flow-oriented way of modelling, the advantage appears in the fact that there are many different processes (flows) throughout the terminal dependent on internal and external conditions, not known at the time of arrival. However, the actions that can be performed by the entities (equipment, terminal personnel, and customers) are known and defined. These two aspects make it easier to model a terminal in an object-oriented way than in a flow-oriented way. Moreover, as control software is also very important to the design of robotized container terminals, the use of an object-orientation as modelling paradigm is rather obvious, since it can be considered as the de facto standard in software development. When the conceptual and simulation modelling is done using a similar paradigm, the gap between functional design and software development may be reduced.

3.4.3 Apply a holistic, layered view on the terminal processes
Our second guideline is to analyse the container terminal from a holistic perspective, taking all processes between the terminal boundaries into consideration. Of course, some processes require more attention than others, but due to many changes, some processes, which at first seem unimportant, may influence the system as a whole. In order to keep the design process manageable, we suggest applying different hierarchical abstractions levels for various system (model) components. Depending on the design activity, we focus on a specific terminal process or component.

3.4.4 Mirror the real system’s architecture into the model’s architecture
It is common to model in accordance with the scope and purpose of the analysis for which the simulation is used. Often, this results in models that are more or less different from the system that will be implemented in reality in terms of structure and processes. That is not a problem in itself; it can even reduce costs of model
development, because the representation of reality in the model was easier to realise and still valid for the purpose for which the model was developed. However, it does not contribute to the re-usability of models within a design project where the same system components are redesigned multiple times. Nor does it support the use of the same models throughout an entire design-engineering process, because there multiple purposes are inherent to the various activities in the design process. Finally, yet importantly, creating a model whose architecture is similar to the real system is beneficial during the implementation process, where it can serve as system environment for functional and technical testing. Therefore, we propose to develop a model suite (see also Figure 3-6) with a similar architecture as the real system, both hardware and software.

3.4.5 Take uncertainty and process variability into account
A dominant property of a container terminal is the lack of deterministic elements, as already has been argued in previous chapters. In order to create a terminal (including its software) that also performs during the operation the design, and in particular the terminal operating system, has to address the dynamic system behaviour of the real system. Therefore, our guideline is to take the dynamic behaviour explicitly into account when modelling and analysing the system, which is principally inherent to a simulation approach. Explicitly modelling the dynamics of processes requires a better knowledge about the range of outcomes of each process, because not only the (estimated) average is required, but also the minimum, maximum, and relative frequency of all possible outcomes are required.

3.4.6 Use the operational processes as leitmotiv for the design
A well-organised operational process is important to a performing container terminal. Therefore, the system design should be determined by the operational processes and not reverse. The knowledge about the operational processes is so important, because the container business is full of exceptions. Moreover, the operational processes cannot be changed easily, because the container terminal under consideration is only a single link in a worldwide network of terminals and changes at a single location could require changes all over the world. However, taking the operational process as leading does not mean that it cannot change, but the boundaries are set, and consequences of changes should be considered in their
An Approach for Designing Robotized Marine container terminals

international context. Examples are, for instance, the existence of a large variety of twist locks used to connect containers, the existence of off-standard containers, or the existence of (local) time-windows for freight transportation, which increase the peak loads at the landside.

However, there are also many processes that can be changed without consequences beyond the particular terminal, maybe except from reactions by the unions. Because most terminals are manually operated, the design of a robotized terminal must consider the actual operational processes, but not blindly automate these processes. Robotization requires rethinking of the operational processes, which includes considering the sensibilities with regard to the involvement of personnel. Many manually operated terminals operate in a similar manner, with similar yard strategies and equipment control strategies that are mostly aimed at limiting the number of additional moves of equipment in order to save on labour costs. However, robotized terminals are different with regard to these particular points, and therefore, reconsideration of operating procedures is required to design an efficient automated terminal.

3.4.7 Integrate the design of manual operations and automated operations

Although many processes are executed automatically at a robotized terminal, manual operations will remain necessary, because of the many physical differences between the equipment in operation (vessels, railcars, and trucks). These differences make it difficult to develop standardised solutions that can be robotized.

Therefore, the interaction between man and machine at execution and control level is a key issue in a terminal’s design. The process operators should understand the way in which the system works - by means of training, but also by means of proper on-line information about the state of the system and the consequences of possible actions - and be able to fully support it. Any interaction between manual operations and automated operations that is not strictly necessary should be avoided. Possibly, incentives might have to be implemented to accomplish this goal. If operators in an automated operation can influence the operation, their interference actions should have a clearly defined functionality and the feedback of their action should be immediate and understandable for operational people.
Therefore, we define as guideline to (1) design the manual operations and automated operations not only from the viewpoint of optimal performance, but also from the viewpoint of optimal controllability by an operator, (2) consider the interactions between the manual operations and automated operations, and (3) involve the (future) operators in the design process, and (4) train the operators to deal with the automated processes.

### 3.4.8 Integrate hardware and software design

A robotized, automated container terminal consists of both hardware (the infrastructure and the equipment) and software (the controls at many different levels, mostly automatically executed though some manual control remains necessary). Usually, these two completely different component types are integrated only after the component design and testing has been carried out. However, the inabilities of one component type can be taken care of by another component type and vice versa. For instance, an advanced AGV routing algorithm can compensate for the lack of space on a terminal site. Alternatively, different curve characteristics of AGVs can solve deadlock problems that may occur in a certain routing algorithm.

In order to find solutions that are not only feasible from both the hard- and software side, but also the best from a technical and economical point of view, integration of both design processes is required. Using a simulation approach can enable this integration, because in a simulation environment, both component types – hardware and software – are represented by software; the hardware as finite state machines, which can be in a number of pre-defined states, and the software in an explicit way by representation of the real algorithms.

### 3.4.9 Make comprehensive and measurable objectives to assess the design

In order to be able to evaluate the design (i.e. the container terminal) afterwards, the objectives have to be clear in advance (Saanen, 1996). The performance indicators should be in line with these objectives in order to be able to determine whether the objectives are accomplished when analysing the performance indicators (see section 2.5). Furthermore, these indicators should be measured on a regular basis during the operation; first in order to evaluate the design when commissioned, afterwards to evaluate the potential of improvements and changes.
3.4.10 Base the decisions within the design process on performance measurements

In order to understand the behaviour of the process that is carried out on a container terminal, adequate measurement criteria have to be developed, because only then relationships between events or actions and the output of the system can be accomplished. In addition, operational data have to be collected in order to determine whether the criteria have been met. The designers can analyse the processes and define the bottlenecks with actual operational information. Therefore, the performance indication instruments should not be limited to the indicators that measure the performance towards the customer. Earlier cases (e.g., ECT-DSL - see section 2.7 - but also Dobner et al., 2002b) have shown that it is better to have too many indicators than to have too few, mainly to create more insight when the system behaves in a different manner than expected.

Moreover, improvements should always be instituted when there is a lack of performance in accordance with the measurement criteria. When these criteria do not converge with both the terminal goals and customer goals, the criteria have not been well defined. Subsequently, the priority of improvements should be determined based on the potential performance increase of the improvement.

3.4.11 Continue monitoring and measuring after commissioning

The environment of a container terminal is ever-changing, for instance the handled volume increases, the size of vessels changes, the modal split changes and the labour cost changes. In most design-engineering processes, the design team is dissolved after commissioning. However, in an ever-changing environment the design process should be continued in order to keep the terminal fulfilling its requirements. In the inductive cases, we have learnt that many terminals do not remain up-to-date in terms of the way to operate and control the terminal when internal and/or external changes take place, which leads to a decreasing service level or a less competitive position because internal and external factors such as labour cost, dwell time and vessel call pattern change.

The (re-)design effort might be at a less intensive level after commissioning, however, the evaluation and improvement process should be continued in order to know whether changes are required or improvements can be made.
4 WAY OF WORKING WHEN DESIGNING ROBOTIZED CONTAINER Terminals

This chapter contains the steps that are typically taken when designing and realising a (robotized) container terminal. In each step, we discuss the main issues, and mention the models that we have developed to support the step. The models themselves are described in more detail in the next chapter.

4.1 A simulation approach throughout the design process

The structure of the way of working is based on the four main activities discerned in the previous chapter, i.e., functional design, technical design, implementation, and commissioning/operations (see Figure 4-1). It incorporates the design guidelines described in chapter 3. Specific attention is paid to the use of a simulation approach in each design activity. Applying this approach throughout the entire design process, leads to a sequence of problem solving cycles (see Figure 4-1). The models that we propose using in the various problem-solving cycles are described together with the specific design issues they support. Although the appearance of the models may differ across the design activities, their architecture remains unchanged (see section 3.4.1). Therefore, the various simulation models form one coherent suite, rather than several stand-alone models.

Figure 4-1: Application of a simulation approach throughout the design-engineering process
Each problem solving cycle consists of the following process steps:

− Creation of a conceptual model. Given the problem definition a conceptual description of the considered system is created (see for definition of a system, section 2.2.1) using an object-oriented world-view. In this context, the problem is to design a container terminal as efficient and effective as possible according to the specific requirements.

− Transforming the conceptual model into a model system, containing all objects and data relevant for the pending design-engineering process. In chapter 5, this process is described in more detail.

− Verification, and validation (these notions will be explained in chapter 5) of the model system. Before starting the analyses, the model system should be a valid representation of the real system, given the scope of the analysis. This means that throughout the design-engineering process, the model system will change because the scope of the analysis changes.

− With the model system, various types of analyses can be carried out, determined by the type of activity (model treatment). Based on these analyses, design decisions can be made or further analyses can be triggered.

In this chapter, the leitmotiv is the sequence of functional and technical design, implementation and commissioning/operations. We do not explicitly follow the steps of the problem solving cycle in the description of the contents of the design process. However, the reader should have this approach in mind as it is the way to make the specific design decisions.

### 4.2 Functional design of a robotized container terminal

#### 4.2.1 Deliverables of functional design activities

In our opinion, the design-engineering process starts with the analysis of the functions the terminal has to fulfil. This question should be answered in relation to:

− The throughput volumes, which the terminal is expected to be able to handle.

− The key terminal parameters, such as the available area, local economic variables, and other site specific conditions.

The second part of the functional design considers the design of the logistic control system, the handling system (e.g. equipment), and the terminal layout. These two clusters of design topics cannot be seen separate from another (see Figure 4-2).
The issues that have to be taken into account answering the questions mentioned above are described in the sections 4.2.2 to 4.2.8. When answering these questions, it is sometimes necessary to go into more detail, already arriving at technical design level (see also Figure 3-5), in order to test the feasibility of a functional design alternative.

The main initial input for the functional design is a market analysis, providing estimates and prognoses with regard to the future volume, the modal split, tariffs, calling patterns, service time demands, customs regulations, et cetera. Usually, a site for the terminal has been chosen when the functional design process starts. This information typically leads to the decision to start an in-depth investigation of how the terminal should look like. In most cases, the uncertainty and unreliability of these estimates and prognoses are rather high and therefore sensitivity analyses on the endogenous variables should be carried out in order to provide a robust conclusion at the end of the functional design.

During the design process, more information becomes available, which usually influences the design activities. The functional design is most directly affected, because the functional design process is most closely related to external factors. Therefore, the starting points for further design activities should be evaluated throughout the entire process, in order to ensure correspondence between the final solution and the starting points of the functional design.

4.2.2 Terminal functionality, container throughput, and terminal performance
The first step in the functional design process of a terminal, especially in the case of new terminal development, is the construction of scenarios for the terminal
throughput volumes. In addition to volume, the type of flows through the terminal (e.g., modal split and transhipment factor) has to be considered. We base these scenarios on economic scenarios, a strengths-weaknesses-opportunities-threats (SWOT) analysis of the new terminal in comparison with competing terminals, as well as on actual negotiations with shipping lines, port authorities, and inland transportation companies. These negotiations can lead to a better balance between service demand and terminal operating costs, especially in the case of a dedicated terminal, which is primarily designed to serve a single shipping line or an alliance of shipping lines.

![Figure 4-3: Example of a volume growth scenario in relation to the required quay length (picture: Bremerhaven CTIII)](image)

The volume scenario should be determined for at least the next 15-25 years, depending on the longest (economical) depreciation time, in order to define a valid business case for new terminal development. In order to create a complete picture, relatively indifferent to variations of economic conditions, it is preferable to define at least three different scenarios that span the range of possible outcomes. Usually a worst – normally the least volume growth –, middle, and best-case scenario are included. In this worst-case scenario, the terminal should still be economically feasible, in terms of profit and payback time.
The terminal throughput potential is determined by the function that the terminal aims to fulfil in the intercontinental transportation network as well as by the geographic location (i.e. the inland infrastructural provisions). The call patterns of a transhipment hub, an import-export terminal, or a regional feeder terminal differ heavily. A transhipment hub does not have much hinterland, which means that most incoming deep-sea or feeder containers also depart via deep-sea or feeder vessels (when transhipment ratio exceeds 0.5, see section 2.5) and that the landside facilities (gate, rail terminal) are of less importance. An import-export terminal mainly serves deep-sea vessels, and containers arrive and leave via continental transportation modes such as short-sea vessels, barges, rail, or trucks. Regional feeder terminals are hardly called by deep-sea vessels; the feeders come from either transhipment hubs or import-export terminals. Therefore, it is recommended to determine which function the terminal will fulfil in the international transportation network. Furthermore, to create useful scenarios, a projection of the future services that will call the terminal, has to be made.

![Figure 4-4: Mutual dependency of a number of key terminal design parameters.](image)

4.2.3 Water- and landside handling capacity, and storage capacity

Based on the scenarios for the throughput volumes, and the function a terminal should fulfil, the main terminal parameters, i.e. the quay length, the waterside (or quay) handling capacity, the yard size, and the landside handling capacity, can all be determined, preferably by means of simulation. Because in many cases the operation of vessels is the key process of a container terminal, the quay length is the most
important terminal dimension. To avoid vessel waiting queues in a port, the only feasible alternative is, in most cases, to increase the quay length and/or increase the waterside handling capacity. The waterside handling capacity is mainly determined by the number of quay cranes, in combination with their gross productivity. Other alternatives are either to alter the contractual service time or the call pattern of the vessels. The service time of vessels is usually very difficult to change (Agerschou et al., 1983) because it affects global sailing schemes, based on the number of vessels per string. At most ports the arrival time of vessels is difficult to control as well. Due to disturbing circumstances, such as weather conditions, delays at other ports, or administrative procedures, actual arrival times may deviate from scheduled arrival times. Therefore, in the analysis we propose to take the dynamics in the arrival pattern (deviations from pro forma berth schedule, as mentioned in Figure 4-4) explicitly into account.

Other issues that immediately affect the vessel service are the waterside handling capacity and the terminal accessibility. These issues are discussed in more detail in the following sections. A final important aspect of the terminal is the yard size. Together with the quay length, this factor has the greatest impact on the physical terminal appearance and the terminal throughput capacity potential. The yard size is also discussed in the following sections.

- **Terminal accessibility**

The general trend in container vessel size (see 1.4) is toward larger vessels. Therefore, the seaside accessibility of terminals is of increasing importance. The accessibility is influenced by the water depth alongside the quay and in the access channel, as well as by tidal movements. Another aspect of accessibility is the sailing time to and from the open sea. Within the Le Havre – Hamburg range there is a large variation in accessibility; some ports require long sailing times (Amsterdam, Antwerp, Hamburg) and others (Rotterdam Maasvlakte, Le Havre, Bremerhaven) nearly none. The longer sailing times have to be compensated by shorter service times or by better hinterland connections, especially when ports serve the same hinterland. On the other hand, the further the vessels move into the hinterland, the more the economy of scale is utilised, since the hinterland transportation is more costly per container than the maritime transportation.
Therefore, terminals should consider their accessibility in relation to their direct competitors and evaluate which service compensation measures have to be taken when accessibility is a comparative disadvantage. These effects should be considered during the functional design.

- **Quay length**

Quay length is normally assessed in increments corresponding to a standard berth, which varies between 250 and 350 metres, depending on the maximum, and average expected vessel sizes at the quay. For a multi-berth terminal it is, in principle, not economical to determine the quay length based on a discrete number of single berths because of the variation in vessel length. This variation is different from port to port, from region to region and from shipping line to shipping line. Therefore, we use the total quay length rather than the number of berths.

An important issue in port operations is the availability of adequate quay capacity. A quay too small will give rise to queues for ships and delays in cargo delivery. Berths that are too small, limit the maximum vessel size, which in turn limits the throughput capacity. Sufficient quay length needs to be provided to enable the anticipated number of vessels to berth with an acceptable level of vessel queuing and berth occupancy. The following factors should be considered:

- Various vessel types and lengths (see Table 4.13).
- Number of containers exchanged per vessel call.
- Distribution of vessel types (calling pattern over the week).
- Vessel regularity (timeliness to ETA).
- Contractual service agreements concerning guaranteed berthing / berth performance.
- Waterside handling capacity.

When determining the optimal quay length, the required terminal service level has to be considered. The most relevant optimisation criteria are (Hillier, 1995):

- Average vessel waiting time before berthing should not exceed more than a certain percentage of the average vessel service time. This is an often used but disputable criterion, because it rewards poorly functioning terminals.
- The minimisation of the sum of idle berth costs and vessel waiting costs.
Ensuring a certain service level. The vessel waiting time should not exceed more than $x$ hours for more than $y$ percent of the vessels served.

The minimisation of total port costs per ton of cargo, in which 'total port cost' is the sum of all terminal costs and all vessel costs.

**Figure 4-5**: Typical relation between berth occupancy and vessel time in port (Dobner et al., 2002b) for a berth of 1,200 m.

In reality, most terminals try to keep the berth occupancy below 60-65% (Saanen, 2002a), keeping a safety margin for early and late arrivals. The berth occupancy is defined as:

**Equation 12: Berth occupancy**

$$BerthOccupancy = \frac{\sum_{i=1}^{n} VesselLength_i \times VesselServiceTime_i}{QuayLength \times MeasurementPeriod}$$

In this formula the berth occupancy is a percentage, and $n$ is the number of vessels in the measurement period, usually one year. The vessel service time is the total time the vessel stayed at the quay, also called the berth time. An alternative measurement method exists that treats a berth as binary object: it is either free or occupied,

---

11 In the vessel length a certain margin for mooring should be included (e.g. 15 meters).
disregarding the vessel length. In that case, the average berth occupancy is higher (up to 80-85%) and the fact that there may be two vessels berth on a single berth – for instance 2 feeders – is not registered.

The longer the quay, the higher the berth occupancy can be, without increasing the vessel service time due to a full quay (Welsing, 2000; Dobner et al., 2001c). The relation between the service time and the berth occupancy is shown in Figure 4-5.

### Table 4-1: Allowable berth occupancy (Welsing, 2000)

<table>
<thead>
<tr>
<th>Number of berths</th>
<th>Allowable berth occupancy (utilisation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 or 2</td>
<td>50 %</td>
</tr>
<tr>
<td>3</td>
<td>60 %</td>
</tr>
<tr>
<td>4 and more</td>
<td>70 %</td>
</tr>
</tbody>
</table>

**Waterside handling capacity**

Besides the quay length, it is the waterside handling capacity that determines the overall terminal throughput capacity as well as the service time of vessels. The waterside handling capacity is the productivity rate with which vessels are discharged and loaded. It is determined by:

- The number of quay cranes assigned to a vessel.
- The technical capabilities of the quay cranes, determined by the technical (kinematic) quay crane characteristics, but also influenced by the vessels that are operated (design and stowage plan).
- The average operational capability of the quay cranes taken all operational losses and/or disturbances (e.g., re-stowage, hatch cover handling, crane movement, meal breaks, crane break down, waiting times for containers due to the yard handling system and/or the horizontal transportation system) into account (gross crane productivity, see also definition in section 2.5.4) (Rijsenbrij, 2001).
- The physically acceptable and attainable number of working hours of quay cranes. The upper limit is determined by the frequency of maintenance and the quay crane structure. The number of working hours that can be achieved is dependent on the vessel arrival pattern.

The waterside handling capacity can only be determined by analysing the quay operation in relation to the horizontal transportation and the yard handling (handling system analysis, see section 4.2.7). In an early stage of the functional design process,
assumptions must be made based on historical productivity data, the type of vessels that will be handled, and the type of quay cranes that are planned. After the detailed analysis of the handling system, the assumptions regarding attainable quay crane productivity, and thus overall waterside handling capacity, should be validated. Eventually, the analysis of terminal throughput capacity has to be performed again when it turns out that the achievable waterside handling capacity is lower than assumed.

- **Yard capacity**

Based on the dwell time, the transhipment percentage, the TEU factor, the terminal throughput (in containers) and the space required for additional services (reefer station, MT-depot, hazardous cargo) the average required yard capacity can be determined. Besides this average, based on static values, there are three other factors:

- Seasonal throughput fluctuation.
- Weekly throughput fluctuation.
- Hourly fluctuation (due to sequence of and imbalance between discharge and loading operations).

In subsequent steps, the roughly determined yard capacity has to be determined in more detail as will be described in section 4.2.7. There the interaction with the selection of a handling system is described. In an early phase of the design process, these details can be taken into account by an area surcharge or by applying simulation, as described in the next section.

- **Landside handling capacity**

The landside handling capacity is determined by three components:

- The gate.
- The rail terminal.
- The landside yard handling system.

Usually, the landside yard handling system has to be able to handle both the peaks from the gate and the rail terminal at the same time.

4.2.4 **Model system of major terminal design parameters**

In order to determine the four main terminal components, quay length, waterside handling capacity, the yard size, and the landside handling capacity including the dynamics in the arrival patterns, the transhipped volume and handling times, we
developed a simulation model, called the Berth Process Simulation model (but see also Ward et al., 2001, Yazdani et al., 2003). This model system comprises the following components:

- Vessel generator determining the arrival time of deep-sea vessels and barges, the vessel length, and the call size (see for instance Table 4-2).
- A quay with a configurable length.
- Quay assignment rules (vessel priorities, position on the quay).
- Quay cranes with a certain gross productivity, determining the time it takes to handle a vessel.
- Additional times for berthing and unberthing.
- Quay crane assignment rules, determining which vessel gets which quay cranes.
- A yard, used for container storage.
- A landside container arrival/departure generator; representing the truck and train arrivals.

![Figure 4-6: Input, system parameters, and output parameters of Berth Process Simulation](image-url)
The relationship between input, system parameters, and output (results) of the Berth Process Simulation is depicted in Figure 4-6. We have applied and validated the model in a number of projects for different terminals (Dobner et al., 2001a, 2001b, 2001c, 2002a, and 2002b).

The run length of the model is one year, the start-up time as long as the longest possible dwell time (normally 20 days), and the minimum required number of replications is six.

Table 4-2: Example of weekly vessel arrival pattern (Dobner et al., 2001c).

<table>
<thead>
<tr>
<th>Class</th>
<th>length (mt)</th>
<th>Call size (TEU)</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barge</td>
<td>110</td>
<td>40</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>Barge</td>
<td>140</td>
<td>65</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Barge</td>
<td>185</td>
<td>80</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Feeder</td>
<td>135</td>
<td>70</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Feeder</td>
<td>185</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Deep-sea</td>
<td>240</td>
<td>300</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Deep-sea</td>
<td>275</td>
<td>550</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Deep-sea</td>
<td>300</td>
<td>700</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Deep-sea</td>
<td>350</td>
<td>1000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>6</td>
<td>11</td>
<td>8</td>
<td>2</td>
<td>57</td>
</tr>
</tbody>
</table>

The analysis using the Berth Process Simulation in combination with the service requirements the terminal has to meet, results in estimates for:

- The required quay length.
- The required number and type of quay cranes.
- The required gross waterside handling capacity (peak).
- The required stack capacity, considering the peak factor.
- The required landside handling capacity (peak).

The way of working (screening and experimentation) using the berth process simulation comprises the following steps, applied to structure the process of determining the key terminal parameters:

1. In the first step, the quay is assumed to be of infinite length, representing a situation where the quay is not a limiting factor to the vessel service. With the infinite quay, experiments with a decreasing number of quay cranes are carried out. Then, the resulting vessel service times are compared with the required service times. This determines then the number and type of quay cranes that are required.
2. In the second step, we continue with the selected number of quay cranes and now we reduce the quay length to determine the required quay length. We analyse the percentage of vessels that can be berthed immediately. Subsequently, we take the vessel types that receive the worst service, i.e. have to wait the most before berthing. From this type we analyse the waiting time distribution and compare it with the requirements, for instance a deep-sea vessel should not wait longer than 8 hours before berthing for more than 1% of the vessel calls.

3. In step three, we check the found combination of quay length and number of quay cranes by analysing the frequency distribution of quay crane use and the quay occupancy rate (measured in occupied quay length x time).

4. Finally, we determine the peak handling productivities that occur during a year and take the 98% value of delivered terminal productivity as measure for the required gross waterside terminal handling capability.

Figure 4-7: Relevant times concerning vessel service (Dobner et al., 2001c)

Preferably, a number of see-throughs have to be made during this analysis. The first see-through has to be made to determine the gross crane productivity. Since it depends on a number of factors, which can change in a new terminal, an analysis of the technical crane productivity and the losses due to operational circumstances (see section 4.2.8) have to be made. This requires in-depth knowledge about the operation. A different approach is also possible: instead of trying to determine the productivity as precisely as possible, one could analyse the sensitivity of this variable and use the variation of the outcome as input for the next step. However, in the
event that completely new crane designs are being considered (e.g. Dobner et al., 2002b), an in-depth analysis is required to estimate in which range the productivity lies.

The second in-depth analysis concerns the berth and crane assignment rules. The knowledge and expertise regarding this process is tacit, meaning that the berth and vessel planners have their knowledge, which is usually tuned to a certain situation (Ugvlig, 2001). A method we used to transform the tacit knowledge into concrete rules applicable to most terminals is by interviewing the experts and by structural validation of our models. Hence, the model does not contain the port-specific or even planner-specific rules, but it provides a sound estimate for the key terminal parameters as has been proved in a number of cases we carried out, where operational people were involved in the validation of the model. The rules that we derived from these interviews have been transformed into if-then-else statements with regard to the number of cranes to use for certain vessels types, depending on the actual situation at the berth and the timeliness of the vessel. For instance, vessels that arrive outside their time-window get fewer cranes assigned if there is a crane shortage.

After the key terminal parameters have been determined, or at least the value range has been limited to a manageable size, the next step, the design of the terminal geometry can start.

4.2.5 Terminal geometry

Often, the terminal layout is constrained by geographic limitations, the presence of other structures or operations, or limitations based on governmental rules. Nevertheless, the common aim is to provide a straight, continuous quay line of sufficient length for the required vessel service, since it provides the least physical constraints and is easiest to extend in the future (see for instance Bremerhaven). A good example is a recently developed terminal in Malaysia (Port of Tanjung Pelepas), where an almost linear extension for 10-15 km of quay is feasible. However, in many cases, a discontinuous quay line is designed on purpose or enforced by the site
conditions. An example is the Ceres-terminal\textsuperscript{12} in Amsterdam. Here, a discontinuous quay line (see Figure 4-8) with a slip dock (or indented berth, Baird and Dougall, 2003) has been designed in order to load or unload a container vessel while quay cranes operate from both sides of the ship. In this way, the number of operating cranes per ship can be increased, which can decrease, in turn, the vessel service times. The future will learn whether this approach is an attractive and beneficial design.

Figure 4-8: Four basic quay geometries (Dobner et al., 2002b)

Other feasible shapes are the corner shape (e.g. Hong Kong) and the peninsular shape. The latter is an interesting one to mention, because we have here a terminal with two parallel quays with the yard in between. This creates opportunity to carry out large transhipment operations without the common long travel distances along the quay causing congestion and delays. When looking at the Delta terminals in Rotterdam, a similar shape can be observed, however the stacking system is separated, as is the control system and stacking equipment.

\textsuperscript{12} The Ceres-terminal in Amsterdam is the first in the world designed to handle containerships with cranes working from both sides. At the ‘Amerikahaven’ a 350 m long slip dock has been realised, equipped with nine post-Panamax cranes - five working from one side, four from the other. All of the 65-tonne capacity twin-lift cranes will be capable of reaching across the full beam of a 6000-TEU ship.
Apart from the external terminal shape, the apron width and the total terminal depth have to be determined. These steps are described below.

- **Apron widths**
  The apron width is the distance from the quay face to the storage area (yard). The distance between the quay crane in its extreme shoreward position and storage area should be as short as possible subject to the following constraints:
  - There should be space for temporary buffer storage of cargo (e.g. for restowage or last-minute feeder cargo);
  - There should be space for lateral transport of cargo to and from open storage areas, and
  - There should be space for access of trucks for direct handling of inbound cargo.

If the apron is immediately adjacent to the ship, the apron width is governed by the distance between the legs of the gantry crane, which is typically 20 to 35 metres for newer cranes and about 15 to 20 m for older cranes, plus the backreach of the crane behind its landside rail, which varies from 10 to 25 metres. Furthermore, the apron consists of the space for horizontal transport between quay cranes and the yard. In Figure 4-9 a cross section of a container terminal is given.

![System boundary](image)

**Figure 4-9:** Cross section of a container terminal (Velsink, 1997).

- **Total widths of land area behind quay face**
  With a crane track of 20 to 35 m, a space for lateral transport of 30-100m (including the backreach of the quay crane) the total width of the land area behind the quay face can be 50 to 140 m (see also summary in Table 4-3). Then the yard and the area for the handling of the continental modes (truck, rail) may take another 300 – 400 m.
Apart from the areas indicated in the cross-section (see Figure 4-9), space has to be reserved for (Velsink, 1997):

- Container freight station
- Workshops for equipment maintenance
- Office buildings for terminal operating company, customs, first aid etc.
- Road vehicle parking
- Entry lanes with weighing and inspection facilities (gate area)
- Off-standard (OOG) containers and hazardous cargoes
- Empty container depot
- Container repair facilities

<table>
<thead>
<tr>
<th>Terminal section</th>
<th>Function</th>
<th>Low [m]</th>
<th>High [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clearance</td>
<td>Service lane to provide access to the vessels</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Crane track</td>
<td>Determined by crane stability, cost and operation</td>
<td>15</td>
<td>35</td>
</tr>
<tr>
<td>Back reach</td>
<td>Space for hatch covers or special containers</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>Traffic lane</td>
<td>A traffic lane for SC, MTS, AGV, Reach stacker</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Storage yard</td>
<td>Stacks for import, export, reefers etc.</td>
<td>Variable</td>
<td>Variable</td>
</tr>
<tr>
<td>Traffic lane</td>
<td>A traffic lane for SC, MTS, AGV, Reach stacker</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Transfer area</td>
<td>Transfer to rail, road or IWT</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Railway tracks</td>
<td>Parking, waiting for hinterland traffic</td>
<td>16</td>
<td>&gt; 16</td>
</tr>
</tbody>
</table>

The design of the terminal geometry is strongly interrelated with the logistical concept and the type of handling system. An integrated analysis by means of simulation can support the decision making regarding these three issues. We stress here the importance of designing the geometry from the perspective of the entire system, avoiding sub optimisation.

4.2.6 Design of logistical control concept

- Typical control concepts

Most terminals in the world have a very simple logistical concept: They operate with gangs and equipment whose work is dedicated to the quay crane, mostly from a single area in the yard. This concept has the following characteristics:

- In the event of a quay crane breakdown, temporary stoppage, or other reasons for delay, the gang dedicated to the quay crane is stopped as well.
- The containers have to be stored at a dedicated location in the yard, which requires a significant percentage of non-productive moves, so-called
housekeeping moves, because it is not always possible upon arrival to store the container there directly, for instance due to incorrect information or long travel distances.

- The reservation of dedicated locations in the yard costs space, since the location is not always used. Therefore, the yard occupation rate is lower.

Especially terminals equipped with RTG and TT work according to the principle described above. A second principle, often used in straddle carrier (SC) terminals, is similar but does not concentrate containers in the yard at a single area. The benefit is that less housekeeping moves\textsuperscript{13} have to be done and that the yard occupation is higher, however, this also has some disadvantages:

- The jobs for straddle carrier drivers are more complex, since they have to drive to a different location each time.

- Tracking and tracing of equipment is required to apply efficient job assignment.

- Most likely, more shuffling has to take place; this means that the container that needs to be loaded is below another container, which has to be moved away first.

In terminals where a high stack density is necessary – e.g. Hong Kong and Singapore, but also Rotterdam and Hamburg – both concepts described above, are less attractive. In dense terminals, either the transportation vehicles do not travel through the yard, or the stacking height is much higher than common in SC-terminals. In such terminals, equipment such as RTGs or RMGs are used, which allow higher stacking and an increased density.

In \textit{automated} terminals two logistical control principles can be used that are not easy to realise in a manually operated terminal:

- The containers can be distributed over the entire yard.

- The equipment can be pooled, i.e. shared between the quay cranes.

The first principle is used to utilise the yard capacity better and is feasible because the housekeeping is much cheaper than in manually operated terminals. The second

\textsuperscript{13} Housekeeping moves are container movements executed in advance of the operation to improve the organisation of the yard. Shuffling moves are moves that have to be executed, because the container that has to be loaded is below another container.
principle is much easier to realise in an automated system than in a manually operated system, because the drivers are replaced by on-board and central computers that direct and guide the equipment through the terminal.

- **Structure of a logistical control system**

When designing the logistic control concept, three layers of control should be discerned (see Figure 4-11).

- Planning.
- Real-time scheduling and co-ordination.
- Execution.

The planning is the process that is executed in advance of the operation (execution). One of the main challenges regarding planning is to cope with the dynamics of the operation and the lack of information (see the sections 3.1.2, 3.1.3, and 0). The following processes should be taken into account in the planning process, in which the vessel arrival times and the load/discharge lists are considered as the input from the shipping lines:

- Berth assignment; determining the berth where the vessel will be moored.
- Crane assignment; determining which cranes are assigned to which vessel for how long.

Apart from the vessel schedules, the input for the berth and crane planning is formed by the current locations of the containers in the yard on the load lists, the required productivity on the vessel, and the final destination of containers that will be discharged from the vessel. All information about vessel arrivals has to be taken into account when determining the berth to moor the vessel, and the cranes that will operate the vessel. Berth planning is done with a horizon of 1 to 2 weeks down to real-time, when the vessel actually arrives. Crane planning is done with a horizon of 24-48 hours down to real-time re-assignment (see Figure 4-10).

The actual container locations are the result of the yard planning, which should be closely related to the berth planning to achieve high berth productivities.

- Yard planning; determining the locations of containers in the yard.

Yard planning is a determining factor for the productivity of the terminal. When many rehandles have to be performed during a peak hour, the performance can seriously drop. Furthermore, when the containers are not stacked close to the berth,
the travel distances get longer, which also influence the productivity in a negative
way. Therefore, the objective of yard planning is to move the containers before the
operation starts or upon arrival at the terminal towards favourable (“pole”) positions,
i.e. close to the vessel and in piles that can be unloaded without shuffling the
containers.

![Diagram of control system layers]

**Figure 4-10: Time horizon of control system layers**

Apart from the planning with a larger time horizon, all processes described above
should also take place on a real-time basis, because that is when most information is
available. In the next stage, a number of other processes should be taken into
account as well:

- Assignment and scheduling of horizontal transportation.
- Assignment and scheduling of yard handling equipment, both for waterside
  and landside service.
- Assignment and scheduling of train and truck handling equipment.

The scheduling and assignment of equipment is a process that takes place real-time
or within a small time-window (up to one hour) before the actual execution starts.
Scheduling (see also Meersmans, 2002) concerns the time when an operation will
take place, assignment concerns the decision which piece of equipment will execute
the operation. The information from the planning forms the input for the scheduling
operation as well as the real-time information about the actual execution. The
assignment is mainly fed by real-time information about the location of vehicles and
the actual scheduled jobs.

After jobs and movements have been scheduled, the execution can start. During
execution, real-time re-planning and movement co-ordination between all resources
should resolve possible deviations from the original plan. Since all processes show dynamic behaviour (see for instance Figure 3-1), regular re-planning – preferably based on deviations from the planning - is required to keep the operation tuned with the schedule.

Figure 4-11: A possible architecture of a logistic control system with multiple layers.

All principles mentioned in this section are reflected in Figure 4-11, which represents a possible architecture of a terminal operating system (or process control system), and represents the architecture of the model that we developed to support the analysis and assessment of the handling system (in this case for a system with automated stacking cranes – ASCs - and AGVs), terminal layout, and logistical
concept. The model is divided in three main layers: planning, scheduling, and execution. Furthermore, the system is organised around the separate subsystems, such as the quay cranes, the yard handling equipment (ASC) and the transportation system (AGVs).

4.2.7 Design of handling system

The third component that completes the design cluster of terminal layout, and the logistical concept, is the handling system, meaning the pool of equipment used on the terminal to transport, tranship, and store the containers.

The selection of equipment used for handling, stacking, and transport of containers, called the handling system, is one of the key issues of the terminal design process. The handling system design has a large influence on the throughput capacity and the layout of the terminal. The key issue here is the balance between handling capacity and storage capacity, because these two factors stand for the terminal’s main functionality. The second key issue in the handling system design is the trade-off between performance (handling and storage capacity) and cost (operational cost and investment cost).

The design of the handling system should interact with the terminal geometry design (layout), because the choice of equipment affects the required space for equipment manoeuvring. The design should also be considered in regard to the interaction with the logistical concept design. Certain types of equipment for instance, are able to decouple subsequent processes, whereas others cannot; see for instance the comparison between AGVs, ShCs (shuttle carriers) and ALVs (automated lifting vehicles) in (Saanen, 2003a). It is also common to use handling equipment that combines processes, such as the straddle carrier.

Furthermore, the design of the handling system depends on the handling equipment that is available on the market or is currently under development. Most terminals worldwide use similar equipment, in a similar set-up. However, with the increasing demands on terminals in terms of handling capacity, stacking density, cost reduction, and environmental control, the current handling systems might not be applicable.

However, the main consideration is the balance between handling capacity, i.e. performance, and storage capacity. These two factors meet at the stack system, which
determines the storage capacity and influences the handling capacity. Therefore, we will first focus on the design of the stack layout and stack handling equipment.

Figure 4-12: Schematic layout of different stacking methods (Hekman, 2001).

- Stack layout and stack handling equipment

Figure 4-12 shows different types of stack layouts, i.e. an all-trailer stack, a forklift truck stack, a reach stacker stack, a straddle carrier stack, an RMG/OBC stack, and an RTG stack. As can be seen from Figure 4-13 there are significant differences between the area utilisation of different yard handling systems. In a time in which the available area becomes scarcer and scarcer, denser yard handling systems as well as higher stacking are becoming more interesting.
### Summary area utilisation of stacking systems

(Based on stacking modules including transport lanes)

<table>
<thead>
<tr>
<th></th>
<th># TEU / ha</th>
<th>Max. stacking heights</th>
<th>Typical average operational filling rate (*)</th>
<th>Recommended max. filling rate in peaks (**)</th>
<th>Average stack capacity (TEU / ha)</th>
<th>Peak stack capacity (TEU / ha)</th>
<th>Throughput capacity per ha / year (***)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reach Stacker, block 3-wide/3-high</td>
<td>258</td>
<td>3</td>
<td>774</td>
<td>55%</td>
<td>85%</td>
<td>426</td>
<td>658</td>
</tr>
<tr>
<td>Straddle Carrier 3-high (1 over 3) (spacing 4.1 m between containers)</td>
<td>265</td>
<td>3</td>
<td>795</td>
<td>60%</td>
<td>80%</td>
<td>477</td>
<td>636</td>
</tr>
<tr>
<td>RTG 6-wide (1 over 4)</td>
<td>268</td>
<td>4</td>
<td>1,072</td>
<td>60%</td>
<td>75%</td>
<td>643</td>
<td>804</td>
</tr>
<tr>
<td>RTG 7-wide (1 over 5)</td>
<td>286</td>
<td>5</td>
<td>1,430</td>
<td>55%</td>
<td>75%</td>
<td>787</td>
<td>1,073</td>
</tr>
<tr>
<td>RTG 9-wide (1 over 4) - Transfer at the end of the module</td>
<td>384</td>
<td>4</td>
<td>1,536</td>
<td>70%</td>
<td>85%</td>
<td>1,075</td>
<td>1,306</td>
</tr>
<tr>
<td>RTG 10-wide (1 over 5)</td>
<td>291</td>
<td>6</td>
<td>1,746</td>
<td>60%</td>
<td>85%</td>
<td>1,048</td>
<td>1,484</td>
</tr>
<tr>
<td>RTG 12-wide (1 over 6) - Transfer parallel to the module</td>
<td>337</td>
<td>5</td>
<td>1,685</td>
<td>65%</td>
<td>85%</td>
<td>1,095</td>
<td>1,432</td>
</tr>
<tr>
<td>OBC 9-wide or MT 10-wide (1 over 4) - Transfer at the end of the module</td>
<td>432</td>
<td>4</td>
<td>1,728</td>
<td>70%</td>
<td>85%</td>
<td>1,210</td>
<td>1,469</td>
</tr>
<tr>
<td>WSG 18-wide (1 over 5) + Buffers alongside 3-wide / 3-high</td>
<td>375</td>
<td>7</td>
<td>2,625</td>
<td>65%</td>
<td>90%</td>
<td>1,706</td>
<td>2,363</td>
</tr>
</tbody>
</table>

**Note:**

(*) The recommended filling rate depends on the stacking strategy and the required workability; when it is higher, productivity losses should be expected

(**) The typical average operational filling rate is the filling as experienced in many terminals

(***) The peakfactor (storage) is assumed to be 1.3

(****) The dwell time is assumed to be 5 days on average

---

**Figure 4-13:** Typical yard densities for various stack handling systems (Rijsentbrij and Wieschemann, 2004)

Besides an increase of stack density, terminals are searching for an increase in performance. Usually these two requirements are not easily to combine, since increased density – measured in TEU/ha – causes a lower accessibility, which is a measurement for handling capacity as accessibility is related to the number of rehandles that have to be performed to load a specific container.

Consider for instance a straddle carrier stack, an RTG stack, and an RMG stack and compare the percentage of jointly accessible containers (see Figure 4-14). In the RMG stack, per stack (in this case 8 TEU long, 6 wide, 4 high, 100% operational filling rate gives 168 TEU\(^{14}\)) two containers can be accessed simultaneously, which is 1%. In an RTG stack, (also 8 TEU long, 4 wide and 4 high, giving a maximum operational filling of 104 TEU\(^{15}\)) each second bay can be accessed simultaneously, which means in this case four containers, i.e. 4%. In the SC stack (8 TEU long, 4 wide and 3 high, 88 TEU maximum operational filling rate), each second row can be accessed on both sides, also resulting in an accessibility of 5%. The longer the stack in the RTG case, and the wider in the SC case, the better the accessibility. However,

---

\(^{14}\) For shuffling purposes at least three container spots should be kept free. Preferably, per bay three slots should be kept free.

\(^{15}\) Because an RTG may not drive with the gantry when loaded, three container spots should be kept free per bay.
each configuration approximately uses the same space, resulting in area utilisation ratios of respectively 1 (RMG) : 0.55 (RTG) : 0.47 (SC).

![Figure 4-14: Three stack handling system; from top to bottom: RMG, RTG, and SC](image)

In the planning of required container yard area, the aim is to provide sufficient space to accommodate the projected annual throughput volume. The required area is normally defined by the number of terminal ground slots (TGS), being the footprint of a standard 20 ft containers (TEU). The necessary number TGS is determined by a number of factors:

- Yearly throughput and transhipment percentage.
- Dwell-time of containers (full, empty, transhipment, water-land, land-land).
- Arrival patterns (vessels, trucks, trains, and barges).
- Peak factor, related to the number of services offered by the terminal, whether it is a multi-user or a dedicated terminal, and related to seasonal and weekly fluctuations.
- Ratio between 20, 40, and 45-feet containers.
- Separated areas (customs, conferences).
– Requirements for special storage (reefers, dangerous cargo (IMO), empties (MTs)).
– The average stacking height of containers and its statistical distribution\textsuperscript{16}. This depends heavily on the information about pick-up moment, port of destination, voyage number, container weight, that is available to the terminal operator.
– The open areas required for operation of equipment and for access to containers.

One simple method that can be used to determine a theoretical measure of the annual yard throughput capacity is the use of the following formula.

\[
C = \frac{L \times H \times W \times K}{D \times F}
\]

– \(C\) is the TEU throughput during a certain time span (in days).
– \(L\) is the number of TEU ground slots.
– \(H\) is the average stacking height in number of containers.
– \(W\) is the average utilisation factor for ground slots.
– \(K\) is the time span (in days).
– \(D\) is the average container storage time in days (dwell time).
– \(F\) is a peaking factor resulting from seasonal fluctuation, weekly variations, and hourly imbalances.

According to Drewry (1998), a peak factor of 1.3 can be considered a representative number, however recent simulation studies we carried out (Dobner et al., 2001c; 2002a) have shown that the weekly and hourly peaks already result in peaks around 1.5. However, the question is what the design value should be, since the absolute yearly peak does not occur very frequently. Therefore, we would take either the 95\% (or the 98\%) peak as design value.

The area requirements for the different stacks (import, export, reefers, MTs etc.) can be written as shown in the following formula:

\textsuperscript{16} The maximum stacking height ranges from one to eight, depending upon the type of yard stacking equipment. The average stacking height as a fraction of the maximum height is governed by the traffic mixture and by the organisation of the yard operations.
\[ L = \frac{C_i \times D_i \times o_i \times p}{365 \times m_i \times h_i} \]

- \( L \) is the area required in m\(^2\).
- \( i \) is the indicator for each container type.
- \( C_i \) is the number of container movements per year per type of stack (regular, reefer, IMO, MT) in TEU (thus the formula for \( C \) should be applied for every container type).
- \( D \) is the average dwell time in [days].
- \( o_i \) is the average space in m\(^2\) per TEU groundslot (TGS).
- \( p \) is the peak factor, which can be treated as one for all container types.
- \( m_i \) is the acceptable maximum stack-filling rate (0.85-0.9)\(^{17}\) that should not be exceeded in say 98% of the times.\(^{18}\)
- \( h_i \) is the average stacking height.

The dwell time has to be considered separately for import / export, transhipment, reefers and empty containers (for which dwell times are usually much longer). The dwell time \( D \) can be written as:

\[ D = \frac{1}{S(t)_{t=0}} \int_{0}^{\infty} S(t) dt \]

In which \( S(t) = \) (quantity of containers still on the terminal)/(total number unloaded containers).

A dwell time distribution according to this formula is given in Figure 4-15. This is the graphical representation of the following formula, which is based on experiences from the ECT home terminal regarding dwell time distribution.

\[
\begin{align*}
S(t) &= 1 \quad 0 < t < 1 \\
S(t) &= \frac{(T-t)^2}{(T-1)^2} \quad 1 < t < T \\
S(t) &= 0 \quad T < t
\end{align*}
\]

\(^{17}\) Higher occupancy rates would lead to the inability to shuffle in the yard, which is inevitable to load the correct containers.

\(^{18}\) It is also to calculate with the average stack filling and a peak factor. In the case this average multiplied by the peak factor than this is the critical value. However, since the berth process simulation has given the 98% value of the peak stack filling, it is easier to take this value.
The factor $T$ represents the time in which 98% of containers has left the terminal. $T$ values up to 10 days for Western countries and this can be 3 to 4 weeks for developing countries. The other 2% can remain for a very long time on the terminal.

![Figure 4-15: Typical dwell time distribution (Velsink, Rakj, 2002)](image)

The factor $m_i$ is related to the so-called workability of the stack. In the event that the stack is fuller than $m_i$, the number of shuffle moves and the duration of shuffle moves increases in a none-linear way, leading to a steep decrease of stack handling productivity.

![Figure 4-16: Empirical dwell time distribution.](image)

In Figure 4-16, a dwell time distribution based on empirical data is given. The same size is over 50 thousand full containers. The average is 5.8 days. As can be seen, this
distribution is close to the distribution described above, but has a higher percentage of containers that stay (very) at the terminal.

- **Gate facilities**

The gate is the entry for the trucks delivering and picking up landside boxes. For import-export terminals, the service to trucks is almost as important as the waterside service. Gate operations are the starting point for export containers and the end point for import containers as far as checking exercised by terminal operators (and of shipping lines as well) is concerned. That is, the liability for the container is in general shifted between terminals and the responsible body for the next transport phase at the gate.

The tasks that must be performed at the gate-area are:

- Documentation (freight documents).
- Visual identification / confirmation (check whether it is the correct container).
- Visual checking of the presence of proper door seals and damage to the container.
- Identification of truck driver.

In the past, it was common to adopt one entrance and exit gatehouse for all containers. The disadvantage of this layout was the risk for congestion, caused by different procedures for different trucks. After all, a truck that leaves the terminal empty requires other procedures than a truck that arrives with a container. Recently, there has been a trend to separate the gatehouse into two or three parts (entrance gate, exit gate, empty container gate, et cetera). This layout is considered very well in avoiding vehicle congestion on access roads to the terminal. However, the drawback is that the capacity is less utilised since it is difficult to convert certain lanes into lanes with a different functionality. This lack in flexibility means that gatehouses destined for truck traffic in the other direction cannot handle peak conditions in truck traffic in one direction. In addition, different gatehouses are sometimes provided for full and empty containers, taking into account the different gate procedures involved.

In order to improve productivity of gate operations, a new system is introduced, called pre-gate. The functionality of the entrance gate is divided into two parts (check and documentation). By means of a simulation – part of the handling system simulation as discussed in section 4.2.8 -, the required number of traffic lanes, the
required parking space before the gate, and the effects of single or multiple gatehouses can be determined. To calculate the required number of traffic lanes, it is necessary to determine the procedures for the various container types.

![Daily truck arrival pattern graph](image)

**Figure 4-17: Typical daily distribution pattern of gate traffic**

In the functional design process of the gate, the major requirement is the truck turn-around time, which is the time between gate-in and gate-out. For instance, in Rotterdam the agreed upon turn-around time is 45 minutes. This turn-around time heavily depends on the number of arriving trucks per hour at the gate. This number varies significantly during the day and week.

For a terminal operator the arrival time of trucks is a black box until the truck actually arrives. This may lead to many additional moves in the stack because it is impossible to pre-plan the yard operations. However, truck appointment systems and pre-advise systems appear now in practice, which allows terminals in preparing the yard for the landside operation as well. This may save yard equipment, because the peak workload decreases, and the yard can be better prepared for the containers that have to be delivered to the trucks.

### 4.2.8 Model system of the handling system, logistical concept and layout

We have seen that handling system, logistical concept, and terminal geometry are the three main components of a terminal. We stated that they should be designed in an
An Approach for Designing Robotized Marine container terminals

*integrated way* and from a *holistic* point of view. The integration and holistic perspective are required to avoid sub-optimisation and unexpected dynamic interactions in a later stage, when the terminal is in operation. The design and assessment of the various handling system possibilities, logistical concepts, and terminal layouts, as well as the analysis of the entire process, should be done using a simulation approach.

When designing the handling system, logistical concept, and terminal layout, we recommend analysing the entire terminal (as model system) under at least four types of circumstances:

1. First, the peak situation in which the most (all) handling equipment is in use and during which the performance is most likely to go down, if the design of the handling system is not up to the requirements. In most terminals, this peak situation does not occur too often, and, therefore, the peak requirement should be defined not taking the absolute peak (in terms of the addition of the maximum waterside and landside terminal productivity), but the 95% or 98% value of the registered terminal productivities. Otherwise, the terminal would have equipment for a peak that seldom occurs, thus creating a low utilisation rate. In such a peak situation, peak flows at water- and landside (trucks, trains) use to clash in time, i.e., a peak on the waterside takes place during rush hour at the landside of the terminals. A realistic peak scenario for both water- and landside operation can be determined by the Berth Process Simulation, described in section 4.2.4. This is for instance a registered terminal productivity, which is only exceeded in 5% or 2% of the cases. Such a scenario may last between 8 and 24 hours, which means that the terminal has to fulfil such peak demands during a longer period. In practice, an even worse scenario may occur, for instance after a large storm. During a storm, vessels cannot enter or leave the terminal, whereas the landside deliveries by rail and road continue. Thus, the yard-filling increases and after the storm, the vessels start entering the terminal and start discharging. Shortly after or even during discharge, the (delayed) landside pick-ups begin. Such a scenario can happen, but in our opinion, in such a situation the offered terminal performance may be lower than the agreed service level.

2. The second situation which is interesting to use as basis for comparing various handling systems, logistical concepts, and layouts, is what we call the ‘average
situation’. This operation represents an operation that is representative for 80% of the possible operational conditions. Typical characteristics are: quay crane usage around 60%, yard filling degree at yearly average and a modest arrival rate at the landside. Because these circumstances occur most of the time, it is a good basis for comparing various handling systems. Of course, the productivity of a handling system should not break-down under peak or disturbed circumstances, but it should excel under average circumstances.

3. The third situation that should be considered is a disturbed situation. Most terminals work well under undisturbed conditions, but whenever unexpected disturbances occur (e.g., breakdowns, software crashes), the performance drops. This is especially the case in an automated terminal, because there, human intervention cannot completely compensate for the problems with the control system. Therefore, all disturbances that could occur have to be investigated and if the effects on the performance are not clear – e.g., AGV breaks down at one of the quay access roads – they should be analysed and prioritised by impact (effect x rate of occurrence).

4. The fourth analysis of the handling system requires a middle-long term perspective, whereas the two previous situations required a short-term analysis. This middle-long term analysis concerns the state of the yard. Since containers possibly remain a long time in the yard, and the fluctuation of the dwell time is high, the effect of peaks in the vessel arrival pattern together with the arrival pattern of landside modes (truck, trains) can create peaks in the yard. During those peaks, when high yard occupancies may occur, it is more difficult to prepare the yard for upcoming operations, which then influences the yard handling performance due to a higher shuffling percentage and longer cycle times of productive moves. Because this is caused by factors that affect the operation (especially the state of the yard) not immediately but possibly over weeks, the analysis period should be stretched to catch this effect. In order to be able to perform this analysis, the simulation should contain a valid depiction of the yard management rules.

The model we use in the design process of handling system (Handling System Simulation), logistical concept, and terminal layout is build according to the structure shown in Figure 4-11. It consists of the following components:
− A quay crane component; representing the cycle time of a quay crane and the interchange process with the horizontal transportation system.

− A horizontal transportation component, e.g. an AGV, a straddle carrier, a shuttle carrier, a multi-trailer-truck, or a tractor-trailer combination. Each component contains a model of the kinematic behaviour of the equipment plus additional subcomponents for collision avoidance or density dependent speed behaviour, interchange operation, order management, and statistics.

− A yard handling equipment component, e.g., an RMG, an RTG, or an OBC. Each component contains a model of the kinematic behaviour of the equipment plus additional subcomponents for collision avoidance, order management, and statistics.

− A stack component, representing a storage module. It contains a database of the containers stored in a module plus the physical properties of a stack module, i.e. length, height, depth. Furthermore, it contains a component to gather statistics.

− Landside traffic components, e.g. trucks or trains. These components contain a representation of the physical behaviour of the truck or train as well as a representation of the load they carry (either full or empty, 1x20 ft, 2x20 ft or 1x40 ft).

− Transfer components, handling the interchange process between quay crane and transportation system, transportation system and yard handling equipment or yard handling equipment and landside traffic.

− A gate component, representing the handling process of trucks when they enter or leave the terminal. This component also contains subcomponents to gather statistics.

− A rail terminal component, representing the handling process of trains. The component consists of the following subcomponents: rail crane, rail track, and train.

− A terminal control component. Besides the control each component contains itself, the simulation contains a representation of the terminal control system. This component is responsible for the tasks mentioned in section 4.2.6. In cooperation with various terminal operators and container terminal software companies, the rules that have been applied are defined, verified, and validated.
The terminal control component is structured according to the three layers mentioned in section 4.2.6, i.e. planning, scheduling and execution control. The planning and scheduling are centralised components, which are organised around the various equipment types. The execution control is distributed throughout the simulation model; each piece of equipment has its own execution control.

In chapter 5, the guidelines applied when developing this simulation environment, will be discussed in more detail. The handling system simulation model is described in section 0.

4.2.9 Design of equipment
A general statement about the design of equipment cannot be made, because it heavily depends on the type of questions to be answered. Our experience is that much effort is put into increasing the technical capabilities of equipment, whereas the operational performance is insufficiently considered. Therefore, we propose to test the equipment within the model system environment when assessing the contribution of increased equipment specifications, or at least test the equipment under realistic conditions. Most of the time, the equipment is idle or waits for another process, especially on a container terminal. Therefore, the focus should be on the utilisation of the technical capabilities rather than on the technical capabilities itself. In order to assess these effects, the level of detail of the equipment model should be relatively high; the movements of equipment, the speed behaviour, the interaction behaviour with other equipment and all process delays have to be depicted. Characteristics such as energy consumption and power supply are sometimes necessary as well to specify the equipment from a system productivity point of view.

4.2.10 Result after functional design
The functional design process should result in a comprehensive specification of all functionalities the system must fulfil, and should contain a conceptual description of all major system components and the way they are interrelated. Based on the functional specification, the operator, or the terminal’s customer, should be able to have a good picture about the service levels the terminal will provide as well as the system that will be realised in terms of quay, cranes, handling system, and the logistic
control concept. Alternatives that have been considered but not selected should also be described, accompanied by the reasons as to why they have not been selected for further development. Moreover, the conclusions of the assessment by means of simulation should be part of the functional specification. It enables technical design teams and implementation teams to review the considerations performed during the functional design.

The functional specification should also contain a description of the sensitivities for major equipment and control components, so that decision-makers can react faster when changes occur during the succeeding activities in the design-engineering process.

The functional specification is also the basis for detailed cost calculations, both for the investment and for the operational costs. Based on the functional specification, an estimate can be made of the investment costs and the cost per move versus the achievable service level, the most important decision criteria for terminal developments or extensions.

4.3 Technical design of a robotized container terminal

4.3.1 Design issues, objectives and decisions

The objective and main result of the technical design is a detailed specification of every component of the system. The level of detail of the specification has to be sufficient for starting-up the production and implementation without (too many) iterations within the technical design. Concerning a container terminal, the following components should be designed in detail:

- Area: ground preparation, under-piling, quay wall, other infrastructure.
- Equipment and buildings: cranes, internal transport equipment, storage equipment, train, truck handling facilities, control room, operations building, maintenance facilities, et cetera.
- Terminal Operating system: planning and control system of each component.
- Communication system: all radio and Ethernet links between control room and ground stations or mobile equipment and between the terminal and external parties (e.g. customers, customs).

As said in the introduction of this chapter, also during the technical design, we propose to apply a simulation approach as main way of working. This means that the
use of models, in particular simulation models, is also important during the technical design process.

Especially during the technical design, when all components are worked out in detail, fragmentation of the design is ready to pounce; more people are involved, and the design team is divided into smaller teams dedicated to certain components. We see this as an inevitable consequence of the size and complexity of the design process, and therefore, we will not touch the organisation of the project in itself. However, the application of a simulation approach for the entire terminal (see guideline 3.4.3, regarding the holistic design perspective) on technical design level can contribute to the avoidance of fragmented design. The way this can be achieved is discussed in the next sections. Another issue of importance during the technical design process is the interaction between the terminal operating system and its future operators. Apart from the involvement of operators and other operational experts in the design process, we discuss a number of possibilities to include them in the experimentation loop of the simulation approach.

4.3.2 A simulation approach during the technical design

During the technical design, a simulation approach can be applied for various design activities. First, it enables the designers to develop and evaluate hard- and software components within the entire environment around a component. This means that various problem-solving cycles – focussed on specific system components – are combined within the model system that is used to compare and assess the various alternatives. For instance, the design of an AGV: what should the properties such as speed, acceleration, deceleration, curve radii, communication time under different circumstances be? Answering these questions in isolation has limited value, because the assessment lacks the effect of the interaction with other components. Consider for instance, the maximum speed of an AGV and its effect on the space utilisation at the terminal. The higher the speed of the AGV, the more space it requires to ensure safe driving; the space required depends on the braking distance of the AGV plus some safety distance (see Figure 4-18 and Figure 4-19). Of course, not only the space utilisation is important, but also the effect on the traffic flows and on the service towards connecting equipment, such as quay cranes and yard cranes. Therefore, we propose testing system components as part of the entire system, so that interactions that will occur during operation, are tested in advance.
Figure 4-18: Space occupation by AGV driving 4 m/s; the red rectangles represent the space reserved for a specific AGV. The AGVs are depicted as black boxes.

Figure 4-19: Space occupation by AGVs driving 7 m/s; in comparison to Figure 4-18, the AGVs need to have more space reserved, because their braking distance has increased due to the higher speed.

In a similar way, the design cycles of the process control system components should be combined. This enables designers to test the software as part of the entire system environment, including the hardware. In comparison with isolated software testing, which is mainly focussed on technical correctness of the software, rather than the functional correctness, the test environment allows for more realistic tests. This has the benefit that various situations (i.e., input and output conditions for the software component or the algorithm) occur as they would occur in a real operation. It also shows the impact on other algorithms, components, or system parts. Take for instance, a job assignment algorithm for AGVs (or terminals trucks). Typically, such an algorithm tries to minimise the driving distance of the AGVs. However, such an algorithm has impact on many other processes and components, such as the impact on the waiting time of other equipment, on the sequence in which jobs have to be executed, on the occupation of transfer points, on the number of driving hours of AGVs, and on the remaining fuel level. Therefore, tasks like job assignment, but also many other tasks or components, should not be analysed in isolation.

A second example is the effect of optimisation. Take again the job assignment of AGVs, and consider the driving times of AGVs. The assignment of jobs might be the result of an optimisation function, however due to the many stochastic effects that occur, the optimisation is per definition based on information that is outdated as soon as the optimisation has been executed. Therefore it is our experience that sound heuristic rules have at least a performance as good as that of an optimisation
algorithm. Besides, the process inside the algorithm is more easily to understand, because by applying the rules, the outcome can be checked by the operator without requiring too much mathematical knowledge. Finally, in most cases, the algorithm based on heuristics is by far outperforming the optimisation algorithm on behalf of the computation speed (see Saanen and Waal, 2001). Especially when these calculations have to be performed in real-time computation speed is an issue of importance.

By means of a simulation approach, it is also possible to prototype solutions for specific components and evaluate them as part of the entire system. Because the model system allows for much easier treatment (scenario setting, replaying specific events, et cetera), and does not always require the complete technical design of the component, this approach may decrease the development time, or allow for evaluating more alternatives.

4.3.3 Results after technical design

The technical design and specifications are the blueprint for implementation, which means that ideally the designers are not involved during the implementation, because the specification speaks for itself. In practice, this is difficult to realise, because of the following reasons:

− During implementation, new design issues pop up, because things appear different than foreseen.

− Components are implemented in a different way, because of a different understanding due to background, and the possibility that an implementation deviates from specifications, although not on purpose.

Therefore, it can be expected that during the implementation, iteration with the technical design, and even with the functional design, is necessary. However, by using a simulation approach as described above, this iteration is likely to be reduced, because already a prototype of most components has been tested on its functional and technical qualities and its behaviour under various circumstances.
4.4 Implementation and realisation

4.4.1 Design issues, objectives and decisions

When all the system components have been designed, the implementation of the software and construction and manufacturing of equipment may start. The objective is to accomplish a realisation of each component that fulfils the technical and functional specification under all conditions in isolation and as part of the entire system.

In the case of software implementation, the border between design and implementation is not as sharp-edged as suggested. Implementation is most likely to start when not everything has been designed in detail. Besides, findings during the implementation may lead to reconsideration of the detailed design. Therefore, the process of design and implementation of the process control software should be suitable for iteration, which again may follow a simulation approach – re-using the model system already developed. This iteration process is eased by component based software development, because a change in one of the components in most cases will not affect other components. The hardware design is less flexible to changes, although here as well, a component-based design allows for flexibility during the production.

The implementation of a design within the context of robotized container terminals concerns in most cases the development and improvement of computer applications, including the automation of equipment. Therefore, high effort is put into the development of the Terminal Operating System in order to make the equipment work in a co-ordinated and efficient way. Usually, the implementation also concerns the installation of new equipment, the training of operators, and adjustment of the terminal layout.

In this section, we will not focus on how to build a quay, or to construct a crane. Nor will we elaborate on how to develop software. We rather want to contribute by describing how we can use a simulation approach during the implementation and testing of the process control system to ensure that the final result meets the specifications. One of the key issues here is testing in order to ensure the working of the system, or a system component under all conditions.

Therefore, the system design should depart from assumption that every component can fail (unexpectedly) in order to create a terminal system that is able to deal with
failures. Every system component can and will fail. The only question is *when* and *how often*. We propose to apply this basic design paradigm to foster the creation of redundancies, failure solving capacity, recoverability, and safety. It is in our opinion, better to have a less performing but robust algorithm than a very high performing one that fails unexpectedly. Also to ensure the seriousness of the handling of disturbances, failure handling, and error solving should not be seen as exceptional processes, but as standard functionality, built in the core of the software.

The application of a simulation approach for testing complex systems is considered necessary because traditional test approaches have a limited richness in their possibilities to create a realistic environment. Offenbacher (2001) mentions five test characteristics that are commonly applicable and point out the limited reach of (traditional) software testing:

- Tests fail to use a representative sample of the actual data to be found in the product’s domain.
- Tests often fail to represent the actual nominal operational environment found in the product’s domain.
- Tests fail to simulate actual maximum load conditions found in the product’s domain.
- Tests fail to simulate actual timing conditions found in the product's domain (especially true in embedded systems).
- Test software is not developed with the same rigor as the production software it is designed to test.

By using a simulation approach, as will be described in the next section, we will try to overcome most of these problems.
4.4.2 A simulation approach during implementation and realisation

A simulation approach during the implementation of a process control system (or Terminal Operating System), consists in our opinion of the following pillars:

- The availability of a model system, consisting of real software components, and models of components. Such an environment provides the opportunity to test components as part of the entire system in a much earlier phase. This environment should have a similar architecture as the real system (see guideline 3.4.4), so that the information exchange between the component that is tested and its system environment is not troublesome. Examples of this on-line connection are shown in Figure 4-20 and Figure 4-21.

- The assessment of hard- and software components on their technical and functional quality. Included in these tests, should be the interactions between components (either as model or real), the delays due to communication, but above all the contribution to system productivity, rather than component productivity.
Creating a test environment that allows plugging in components is likely to reduce the duration of the feedback loop towards technical design and accomplished implementation work at component level, because it creates test facilities in an earlier stage of the implementation process. It also increases the realism of the tests; isolated tests, although useful, often do not represent the specific interactions that occur in an integrated system. Furthermore, isolated tests only assess the functional performance of a component to a small extent; they mainly assess the technical performance, in terms of response times and input-output relations. Lastly, when the real software is integrated, the quality of the individual components is likely to be higher, because many integration tests have already been carried out using the model system as environment for the components.

Figure 4-21: Example (2) of a production software component tested as part the model system environment. All components in grey are in real mode; all in blue are in simulated mode.

In order to be able to apply the model system for the purposes mentioned above, a number of measures have to be taken. The main measure is mentioned as guideline 3.4.4, stating that the architecture of the model system should mirror the real system’s architecture, including all hard- and software. In practice, this means that the communication between all system components have to follow the same patterns in the model system as in the real system. As is shown in Figure 4-20 and Figure
the links between model system and real system components remain unchanged when exchanging models by real components. Current technology does support this approach; via the TCP/IP or CORBA protocols, a number of simulation packages can exchange such messages. Besides the structure of the model system, it is important to establish a close co-operation between the developers of the software and hardware components and the designers applying the simulation approach. Otherwise, either the model system does not represent the most recent functionality of the software or hardware, or the interfaces between the components do not correspond to the interfaces in the real system anymore.

For people at management level it is often difficult to comprehend the detailed technical process, due to a lack of technical background and/or due to the lack of time to get deeply involved in the system. However, the manager’s task is to continuously evaluate whether the design teams are heading towards the solution that fulfils the requirements. For this purpose, applying a simulation approach is key to provide this insight throughout the design-engineering process. First, because the experiments result in quantitative data, which present the system behaviour in terms of the performance indicators, which should be directly derived from the system’s requirements. Secondly, because the visualisation may create a better understanding of the functionality the current implementation provides. Finally, the comparison of various alternatives provides information about the sensitivity of the system for changes and may establish more confidence in the selected design(s).

Besides, the model system should be used also as a test tool for checking whether the components provide the required functionality. In such a way, an extra functionality check is built into the development process. These tests can not only take place when the component is designed at an aggregate level, but also in a further stage in the design process, for instance when a prototype has been implemented, or when the finalized component is in the pre-operation phase. When the model systems architecture equals the real system’s architecture, and the interfaces are well specified, it is possible to test system components in these various stages; both on technical stability and on functional performance.
4.5 Commissioning and Operations

4.5.1 Design issues, objectives and decisions

When a system or solution has been implemented, the commissioning, i.e. the acceptance by the user, can begin. The commissioning can last more than a year, but is commonly planned one to three months. After the system has been accepted by the user, the ‘real’ operation can begin. In the examples of automated terminals yet, the terminal was not immediately at full capacity during this first period. It took ECT more than a year (1992-1993) to get the terminal working at an acceptable performance. At CTA\(^{19}\), there have been some start-up problems as well, which have affected the system’s performance. These problems do not only occur in exception situations, but also during regular operations.

After the first period of operation, the system will get more stable and problems are most likely to occur in exceptional situations that have not been expected, or due to malfunctioning equipment, unbalanced performances, unknown procedures, new communication lines, new tasks and responsibilities et cetera. Therefore, the start-up of operations includes much fine-tuning, training, and optimisation activities. Usually, after the first year, the system works according to the design specifications. To prove this, an ex-post evaluation – performance monitoring – should be carried out. Only then can be assessed whether the objectives – in terms of fulfilment of performance levels – have been achieved and against which cost. From then, a new design-engineering process will most likely start, trying – again – to close the gap between targeted performance levels and actual performance levels.

Although the system might perform according to the design specifications, it does not mean that the (re-) design work is over. In order to keep systems like a robotized marine container terminal performing well, continuous monitoring and improvement are likely to be necessary. In an automated system, it is important to keep the software up-to-date, especially when the computer hardware or the equipment changes. Because it is still the case that computer power is the limiting factor for some computerised processes, like optimisation algorithms, it is likely that during the development, compromises have had to be made in order to achieve an operational performance.

\(^{19}\) Container Terminal Altenwerder in Hamburg.
system. However, in the near future computational boundaries will most likely be extended, so updates can and should be made to realise further improvements.

Furthermore, during the operation, the best course of action is not always clear to the operators. Various external influences – late (clashing) vessel arrivals, landside peaks, last-minute changes but also bad weather, strikes, and serious equipment breakdowns – imply unanticipated situations. The decisions that then have to be taken in real-time, have effect on the operations in the next shift or even longer.

Therefore, we adhere to a simulation approach during commissioning and the operation as well. In the next section, we focus on the way this can be accomplished.

4.5.2 *A simulation approach during commissioning and operation*

The model system used during functional and technical design and the implementation activities can be applied during commissioning and operations, because as a consequence of the on-line testing with real software, it has a sufficient level of detail, and according to guideline 3.4.4 the architecture matches the real system’s architecture.

When the system – both control system and equipment – is completely built, a lot of fine-tuning has to be done, in which a simulation approach fits well. Instead of trying the changes in the real system (trial and error), – especially when the control system is concerned – the simulation can be used to test the adjusted algorithms or procedures, which is less costly and less disturbing the ongoing operation. This is possible because the simulation models, in the case they have been developed and applied as described in the preceding sections, have the same structure and interfaces as the real control system. This approach enables the following activities:

− Testing of changes under manageable conditions, so that problem situations can be replayed.
− Testing of changed components under a wide range of conditions, without disturbing the live operation. Before the improvement measure is applied in the real system, it can be tested off-line, linked to the model system.
− Testing of changed components during a long period of time. Because commonly a model system can run faster than real-time, longer operations can be analysed, without delaying the entire commissioning process, or without letting the problem in the operation remain.
− Analysing problems that have occurred. When the model system can be fed with the data log of the real control system, situations can be replayed relatively easy. Then, one can analyse which alternative strategies might have been more successful.

Furthermore, the model system environment can be used for on-line decision support. During the operation the model system can be used as a tool to anticipate problems (e.g., deadlocks, planning unbalances, failures, resource shortages). During the operation, the model system should be adjusted and tuned until it is a valid representation of the running system. When this has been accomplished, it is possible to run the model system with input data from the real system and continuously wind the tape of events forward, to be able to anticipate on problems. If problems occur in the model system, there is still time to take action before the real system runs into the same problematic situation. Furthermore, the model system can be used for finding deviations between model system and real system, which in most cases will point out individual equipment problems (failures, wear, or exceptions). When the simulated equipment is tuned well, these deviations should be noticeable.

A third application of the model system is the use for demonstration and training. One can more easily show what the impact could be of bad decisions, a lack of information, or an inadequate maintenance program. Besides, operators can be trained to operate and monitor the system under all kind of circumstances without interfering with the real ongoing processes, whereas the interfaces and behaviour of the system are similar. We experienced that the tasks of operators of an automated system are not always easy because of the complexity of the system, and the long feedback time, i.e. when an operator changes a parameter, the effect on the system can only be measured over a longer time, whereas the operator tends to assess the direct impact of a change.

The fourth application, for which the model system can be used, is to support the daily planning. By using the model system, the berth assignment planning, the crane assignment planning, and the yard planning can be simulated in advance to determine what kind of problems could occur. One could go even one step beyond by optimising the planning by means of a simulation approach (Sonneville, 2002).
5 **Way of Modelling when Designing Robotized Container Terminals**

In this chapter we elaborate on the models and the approach followed when designing and developing the models. Here we take advantage of the building block based approach taken as one of our starting points. Also a number of other guidelines mentioned in chapter 3 are applied in the model suite.

### 5.1 Overview of simulation models

In the previous chapter, we elaborated on the contents of the design-engineering process and described how a simulation approach can be applied throughout the process. As part of the approach, we already mentioned the suite of models that is used. In this chapter, we elaborate on this suite of models that we have developed as part of our proposed simulation approach. First, we discuss the concepts behind the structure of the model suite, as well as the structure of the model components (section 5.2). Then, we discuss each specific model being part of the suite one by one. These models are aimed at answering different design questions, which may be relevant throughout the entire design-engineering process. Three models are described (section 5.3):

- The berth process simulation, focused at the key terminal design parameters.
- The handling system simulation, aimed at the balanced design and selection of terminal components – hardware and software – that are present at a terminal.
- The equipment simulation(s), aimed at design of specific pieces of equipment.

Finally, we discuss the coherence between the three types of models (section 5.4).

### 5.2 Architecture of the model suite for container terminal design

Since we propose to use the simulation approach throughout the entire design-engineering process, it is beneficial to **re-use** the simulation models throughout the process. This re-use is beneficial because it would take much more time to develop new models for every single activity of the process. Especially during the technical design and the implementation, the required level of detail in the models leads to a significant lead-time of the model development. The re-use has the additional advantage that it saves the verification and validation of the re-used components.
In order to create a model suite allowing re-use, the architecture of the suite should match the real system’s architecture. This puts additional requirements to the architecture of the models, because during the various activities of the design-engineering process, the models are used for different purposes, in different contexts, and at varying levels of detail.

5.2.1 Functional structure

The first decision concerning the model architecture was to apply an object-oriented approach (see guideline 3.4.2). This approach already determines – at least to a large extent – the way of transferring real-life phenomena into a model. Secondly, we decided to structure according to the function of objects (Hall, 1989). A functional structure has been applied at building block level, i.e. at the level of compound objects, such as an AGV, an RMG, a quay crane, or specific parts of the process control system. A functional structure has also been applied to so-called component building blocks, smaller objects out of which compound building blocks are created (see Figure 5-1). Compound building blocks are of a heterogeneous nature with regard to function, whereas component building blocks fulfil specific functions, such as the representation of kinematic behaviour, the visualisation of compound building blocks, or the gathering of statistical data. As most properties of these component building blocks are similar for various compound building blocks, they originate from the same mother object (see Figure 5-1).

According to Hall (1989), there are three other options, besides a component’s function, for creating a component structure:

− Time structuring.
− Structuring by geographical division.
− Structuring by minimising interactions.

Time structuring does not make sense in this domain, since the relevant processes at container terminals take place at the same time. Neither do they follow a standardised sequence in time, which makes structuring in time difficult if not impossible.

Secondly, structuring by geographical division is also impractical, since a container terminal is bounded to a limited and clearly definable location.
The third option, on the other hand, is one to consider, since interaction is a possible source of disturbance at a terminal. Furthermore, the less interaction that exists between system components, the easier components can be exchanged without affecting other components. This latter point is interesting from a software engineering point of view as well as from the perspective of using the same simulation models throughout the design-engineering process. The easier components may be exchanged, the easier more detailed components can be plugged into the already existing simulation environment. Especially when designing and implementing the terminal control software, the simulation remains valuable, because ready software components can be tested as part of a simulated system.

![Diagram showing functional decomposition in multiple object trees](image)

Figure 5-1: Inheritance structure within the simulation suite. Functional structured component trees deliver components that are combined to building blocks, which are instantiated in the model (Verbraeck et al. 2001).

However, since it is common in software-engineering to structure according to the function of objects, we follow this approach, because it is likely to fit well with the structure of the process control system. Furthermore, a functionally structured model fits well to the objective of reducing interaction between components because a
sound functional structure should avoid cross-functional interactions. Therefore, we have chosen for a functionally structured simulation library (see also Verbraeck et al., 2001).

5.2.2 Polymorphism
A useful feature of object-orientation is the use of polymorphism, which means that objects with the same functionality and interface can have multiple implementations. When we project polymorphism onto the objective to use the same simulation models throughout the entire design-engineering process, we can conclude that this property is well usable to apply detailed and aggregated implementations of different components in one model, depending on scope and purpose for which the model is used. Reynolds and Natrajan (1997) address this issue as cross-resolution modelling (CRM). The choice of a certain implementation depends on the state of development the component has. It might just exist as a rough design, as a prototype or even as ready production software. For the other system components, this should not have any impact, because the interface with the other components remains unchanged.

5.2.3 Three layers of control
Additional to functionally structured models, we have applied three layers of control (see Figure 5-2). The highest layer of control comprises functionality like fleet management, job assignment, yard management, central database, planning and scheduling (see also Figure 4-11 and Figure 5-10). In principle, all functionality that exceeds the individual piece of equipment is located at fleet manager layer. Therefore, we classify the fleet manager layer as a centralised control unit, whereas the equipment control and motion control layer are decentralised control units. The second layer we call the equipment control layer. This control unit provides the functionality of routing of single units, such as an AGV, a quay crane or a stacking crane. The equipment control is responsible for taking care of an individual piece of equipment, making sure that is executes its jobs in-time, only drives in a safe area, et cetera. It has a one-to-one relation with a piece of equipment. The motion control is responsible for carrying out the physical movement(s). It controls speed and steering of the equipment, but also registers for instance the driven distance.
Figure 5-2: Three layered structure of simulation model suite; between the layers messages are exchanged in an asynchronous way.

5.2.4 Asynchronous communication between components
The basic communication principle applied in the model suite, is the asynchronous command-event mechanism (see Figure 5-2). This mechanism is especially used for communication between the various control layers. A higher layer control unit gives a command to a lower layer control unit, e.g. the AGV-MS – RouteManager sends a route to the AGV-RouteControl for execution. This statement is sent like:

\[ \text{AGV-C(13).RouteControl.ExecuteRoute(ID, Destination, Due time)} \]

After receiving the route, the AGV-RouteControl sends a confirmation of receiving, such as:

\[ \text{AGV-MS.RouteManager.RouteReceived(ID)} \]

Then, the AGV-RouteControl translates the route into movements from co-ordinate to co-ordinate that the Motion Control layer can translate into triggers for the (simulated) engine and steering. As soon as the movement has been carried out completely, the AGV-RouteControl sends an event to the AGV-MS – RouteManager that confirms the completion of the route, using a statement like:

\[ \text{AGV-MS.RouteManager.RouteCompleted(ID)} \]

During this route execution – and until the AGV-Control has sent a notification of receipt -, the AGV-MS is not waiting for an answer; the route has the state “in
execution” and until the specific AGV-Control sends the message to confirm the completion, the AGV does not get another route.

The application of asynchronous communication is important to decrease the dependencies between the implementations of components. Dependencies reduce the flexibility during the component design and redesign process that will take place during the technical design and implementation.

5.3 Model suite for robotized container terminal design

5.3.1 Overview of models

As we already said in the introduction of this chapter, the model suite consists of models at three levels of detail.

- The berth process simulation, focusing at the key terminal design parameters.
- The terminal handling system simulation aiming at the balanced design and selection of terminal components – both hardware and software.
- The equipment simulation(s), aiming at design of a specific piece of equipment.

The three models are aimed at supporting different decision-making processes. In the Table 5-1, we summarise the main questions that are addressed by these models:

*Table 5-1: Typical questions addressed by one of the models in the simulation suite*

<table>
<thead>
<tr>
<th>Typical decision-making issues</th>
<th>BPS(^{20})</th>
<th>HSS</th>
<th>ESs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determining a long-term throughput development path for a terminal, including the required quay length, stack capacity, and quay handling capacity.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluating the throughput berth capacity of a wharf given certain vessel call patterns, berth length, number of cranes, crane productivity, and acceptable probability of a ship waiting for a berth.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluating the maximum expected container throughput from a berth into a container yard.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluating the container yard storage and throughput capacity.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Evaluating the throughput capacity of a gate to a container terminal.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluating the throughput capacity of a rail facility within or connected to a container terminal.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluating a terminal from a holistic perspective, i.e. including the interaction between the quay, the yard, the gate, and the rail interfaces.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantifying the effect of different layouts, equipment</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{20}\) BPS = Berth Process Simulation; HSS = Handling System Simulation; ESs = Equipment Simulation(s).
Chapter 5: Way of Modelling

5.3.2 Berth Process Simulation

The long-term development of a terminal mainly concerns quay length, stack capacity, and waterside handling capacity, i.e. number of quay cranes and quay crane specification, and landside handling capacity, i.e. gate capacity and rail handling capacity. All other system variables are derived from these key parameters (see also 4.2.3). In order to determine these key parameters we use a rather abstract model, because the focus is on long-term development, not on short-term behaviour of the handling system. This means that depiction of the handling system can be on an aggregated level. Although we chose to implement the model in an object-oriented programming language, the dominant modelling paradigm is flow-orientation; the vessels are followed through the processes that are executed (see Figure 5-3).

![Figure 5-3: Process flow in berth process simulation (see also Figure 4-7).](image)

In the berth process simulation, the following reductions are applied:

- The gross quay crane productivity over a longer time is assumed constant. This means that disturbances at micro level – due to breakdown, meal-breaks, and operational slow-downs – are included in the gross crane productivity (see also Figure 2-4). This means that the assumed gross crane productivity contains all
delays caused by the transportation and yard handling system. In practice, this definition is easy to accomplish, since terminal usually measure the gross productivity in a similar way.

- The stack capacity is considered unlimited and the effects of a fuller stack on the productivity of the quay crane is not taken into account. See also the previous explanation.

![Figure 5-4: Main components berth process simulation](image)

The time horizon applied is usually one year. During this year, 52 weeks with a similar pro forma berth schedule are simulated. However, the actual arrival time of a vessel varies per week, as does the call size (see Figure 5-4). Each individual container is picked-up or brought by either rail or a truck, which has an arrival time that is in accordance with the dwell time distribution (e.g., Figure 4-15) and in accordance with the gate and rail service hours (e.g., Figure 4-17).

The individual vessels are berthed on the quay, as close as possible to their planned berth location. If the quay is fully occupied, the vessel waits at the anchorage point. After being berthed, the vessel is operated by a number of quay cranes, in accordance with the required productivity on that type of vessel. When less quay cranes are available, it may happen that during the operation one or more additional quay cranes are assigned to this vessel to be able to catch up. However, the maximum number of cranes to be assigned to a vessel is limited by the number of bays, and possible other physical constraints (e.g., size of a quay crane, vessel length). This process results in a quay occupancy scheme as shown in Figure 5-5. Here the actual position of the vessel on the quay, the planned time of arrival (ETA) the actual time of arrival (RTA), the actual call size, and the quay cranes that handled the vessel in time are visualised. On the horizontal axis the berth length is shown; on the vertical axis the time in hours. Here, a little over four days is shown.
5.3.3 Handling system simulation

In section 4.2.8, the model system for supporting the selection of an appropriate handling system has been introduced and the main components have been briefly mentioned. As has been described in the previous section (see Table 5-1), the handling system simulation is aimed at designing all terminal components – hard- and software – as an integrated balanced system. The questions focus on the selection and dimensioning of the handling system, i.e. transhipment, transportation, and yard handling equipment. However, the model comprises of more components, such as a rail terminal (using the train generator from the berth process simulation, or alternatively a detailed rail terminal model), a gate (using the truck generator from the berth process simulation, or alternatively a detailed gate model), and a terminal control component, covering most functionality required to control a robotized container terminal. An example of the component structure of the handling system simulation is depicted in Figure 5-10, although this example does not cover the landside handling system.
There are two purposes for which the handling system simulation can be used to test the real software of a process control system (PCS). The first type of test concerns single, individual software components of which the functionality or the technical quality have to be tested. Usually this takes place when the entire PCS is not yet ready for integration testing. The second type of test is when the entire PCS is ready, but the equipment is not yet installed, or testing with real equipment would create too much hinder for the operation.

In the first case, a number of provisions have to be made to be able to test specific, isolated control functionalities. These provisions mainly concern specialisation steps and decomposition steps, because the interfaces at component level have to match with the interfaces of the real PCS if the goal is to include PCS components to the model system. Since the main benefit of applying a simulation approach for testing PCS software is reached when only those components are tested that have a major impact on the performance, the focus should be on testing these specific components.
In the second case, only a part of the handling system simulation is required. This model only depicts the equipment, the on-board equipment controls, and the interfaces to the PCS. It depends on the functionality of the PCS, what kind of reductions can be made within the model of the equipment. When for instance, the PCS contains functionality for planned and scheduled driving of AGVs (Verbraeck and Saanen, 2003), then the PCS contains a model of the AGV. In such a case, the model of the AGV should be accurate enough to test the functionality of planned driving. Usually, a good depiction of the kinematics under regular conditions is sufficient, but it might also be required to have failure behaviour implemented.

![Operational productivity = 35 ccph](figure)

*Figure 5-7: Example of cycle time distribution of a single hoist quay crane (Saanen et al., 2003)*

To arrive at a workable but valid model, we have applied a number of reduction steps:

- The run length of the model is less than 24 hours, with one hour start-up time.
- The cycle time distribution of quay cranes is according to an empirical distribution, as shown in Figure 5-7. This means that the correlation between the duration of consecutive moves is assumed to be inexistent.
- The initial stack configuration is assumed to be according to a number of rules, as it cannot be built up during the run, as this takes at least a run length of 20 days. Therefore, the origin of containers, the compound of container piles, the filling rate, are based on assumptions that are in line with a specific scenario.
Of course, the consequences of these assumptions can be analysed by means of sensitivity analysis.

- The behaviour of equipment is modelled according to the equipment specifications. This means that for instance a speed variation – that occurs in practice – is not taken into account. However, the model of vehicle kinematics is detailed, including start-up time, reaction time, communication time, acceleration, deceleration, and varying speed depending on the type of movement.

- Information is assumed to be perfect; containers are not lost in the stack, load lists are considered perfect (no over landed or short landed containers).

- All containers on the load list are present on the terminal, which means that late arrivals are not considered.

![Figure 5-8: Screen shot from the handling system simulation of a straddle carrier terminal.](image)

With regard to the model of the process control system, we found in earlier analyses (see Dobner et al., 2002a, 2002b, 2003a, 2003b) that a comparison between different equipment types can only be made when the representation of the operation is valid, which is heavily influenced by the algorithms, procedures and business rules within the process control system. Therefore, we considered the following aspects:

- The arrival sequence of containers under the quay crane during a loading operation: in most cases, this sequence has to be kept in a very strict way.

- The distribution of container in the yard; depending on the type of operation, there are specific rules that determine this distribution.
The dispatching rules for container handling equipment; in manually operated terminal most terminal apply dedicated teams to the quay crane. In an automated environment, the sharing of resources between points of work is common, which requires algorithms like the Hungarian algorithm to assign equipment in an efficient way.

- Routing of transportation equipment.
- Collision avoidance between automated equipment.

Other operational conditions that have to be considered because of their impact on the assessment of various equipment types are:

- Twin-lifting / twin-carry operations; quay cranes performing a loading operation, loading two 20 feet containers in a single move.
- Dual cycling of quay cranes, which means a combination of discharge and loading moves.

![Screen shot from the quay crane model that contains all kinematic movements of the machines present in a quay crane. Shown is a real vessel operation with tandem-lift operation. The quay crane is a dual-trolley quay crane. The handling system is a straddle carrier operation.](image)

The architecture of the model system is depicted in Figure 5-10. During the design of this model, the following modelling guidelines were followed:
The model system has been designed according to the rules of object orientation.

The model system consists of three layers, which we address as fleet management (comprising of the major planning components), equipment control, and motion control (see section 5.2.3).

The communication between the layers within the model system is asynchronous. This is done, because in practice these processes are also carried out on multiple machines, and therefore in an asynchronous way.

The relation between layers can be classified as a command-event relation, which means that a task is sent to a lower level (command). The lower layer sends back an event when the task is performed. In between, the higher layer does not wait for the lower layer to answer.

The entire architecture is structured around the components that fulfil certain functions. This means that the system operational at the terminal is represented in a similar way in the model (and, following from design guideline 3.4.4 this is also the case with the process control system).

Within functional components, a further decomposition into subcomponents is used, again around functionalities. Take for instance the horizontal transportation system, which consists of a router, a collision manager, a deadlock avoidance mechanism, an order manager, and a tracking and tracing component.

The multi-level architecture enables the modeller to exchange certain components relatively easy – because the functional components remain unchanged when for instance the horizontal transportation system with terminal trucks is replaced by AGVs – and allows for more or less details per functional component.
An Approach for Designing Robotized Marine container terminals

Chapter 5: Way of Modelling

![Architecture simulation model for analysing handling systems](image)

Figure 5-10: Architecture simulation model for analysing handling systems
5.3.4 Equipment simulation

As part of our model suite, we developed the following equipment models that can be used as stand-alone components, to tune the technical specifications of the equipment:

- A quay crane model, containing most types of quay cranes available on the market, plus some new designs that are researched (see Figure 5-9).
- A mobile harbour crane model, representing a level-luffing slewing crane.
- A stacking crane model, depicting either a Rail Mounted Gantry, a Rubber Tyred Gantry, or an Overhead Bridge Crane (see section 2.3).

![Figure 5-11: Screen shots form RMG simulation (left) and QC simulation (right)](image)

The equipment models have the following properties:

- All movements are computed in three dimensions and in time.
- The hoist movement is weight dependent, i.e. in the case of varying container weight the speed and energy use of the movement is adjusted.
- All dead times within a movement are considered.
- The energy use and power requirement of each component is considered.
- The possible wind influences are considered.
- In the case of the quay crane model, the forces on the crane legs are considered.

5.4 Model coherence

The application of a simulation approach throughout the design-engineering process benefits from coherence between the models. As has been described in the previous sections, for the different design issues, different models can be applied. Furthermore, we discussed the use of the same model for various purposes, i.e.,
assessment of alternatives, prototyping of algorithms and testing of software components. So on the one hand, we have a suite of models that each serve a different purpose throughout the design process, on the other hand we have single models serving multiple purposes, which are thus re-used throughout the design process.

In the first situation, we apply models that are linked via so-called input-output relations, i.e., the output of the one model is used as the input for another model and vice versa: the output of one model is used to validate assumptions made in another model. This situation applies mostly when we have models of a different level of detail; in the less detailed models, assumptions are made, which can be validated by the more detailed models. On the other hand, the more detailed models usually have a more limited time-horizon and therefore the input, for instance an operational scenario, is generated by the less detailed model, using a larger time-horizon. In the model suite, the following input-output linkages are used:

− The output of the berth process simulation (see section 4.2.4 and 5.3.2) is used as input for the handling system simulation. In particular, the berth process simulation results in operational peak scenarios, i.e., the maximum (gross) berth productivity delivered on the waterside, the maximum truck arrival peak at the landside, the maximum yard occupation, et cetera, that can be applied in the handling system simulation (see section 4.2.8 and 0).

− The handling system simulation validates the assumption of quay crane gross productivity that is made in the berth process simulation. Because in the latter model, it is assumed that this performance can be maintained under all circumstances, the handling system simulation checks this under the worst circumstances. If this assumed performance proves to be unachievable, the berth allocation simulation has to be performed again with adjusted input values.

− The equipment simulation models provide input for the handling system simulation as well; because the equipment simulation models are – in some cases - too detailed to be used in the handling system simulation. This is done e.g. for the cycle time distribution of the quay crane.

In the second situation, we use the same model for multiple support purposes as has been shown in section 4.4 and further. Here, the model components are used in a
blend with the production software or equipment. By doing so, the test environment for the production software is similar to the environment with which the components have been prototyped and evaluated on performance.

5.5 Visualisation

When using a simulation approach, an important role is played by the visualisation of the models. As humans are better at processing and interpreting graphical data than numerical information (Kasanen et al., 1991), the visualised processes are mostly better understood than the functional description of complicated algorithms. In research it has been proven that graphical methods – for instance used in decision support tools – for representing information, systems or alternative solutions, improve performance, understanding, and solution quality for users (Pirkul et al., 1999; Larkin and Simon, 1987; Carrol et al., 1980). These observations stress the importance of an easily interpretable visualisation of complex systems as counterpart of a clear, written, description.

![Figure 5-12: Two different visualisations of the same process. The 3D view of a mobile harbour crane model (left), and the 2D view of the same model (right). In the 2D model, the exact movement of the crane can much better be validated than in the 3D model, because the trajectory of the spreader can be followed more easily.](image)

This is probably the reason that simulation packages increasingly include advanced visualisations. Also fostered by the increased technical capabilities of hard- and software, most common-off-the-shelf simulation packages include the possibility of 3D visualisation and true-to-scale 2D visualisation. This makes it possible to create an environment, which is quite recognisable for operational experts and decision-makers, who have difficulties in understanding symbolic or abstract representations. Advanced visualisation also has a major disadvantage: it conceals the real functioning
of the model, because people focus too much on the visualisation, which does not show the entire operation in the right perspective. The modeller should keep this in mind during the validation of the model.

A visualisation also works as a vehicle for communication. During a multi-disciplinary project as the design of a container terminal, it is difficult to keep everyone on track and to keep the system components aligned. We have experienced this during a two-year design project of a fully automated transportation system at Amsterdam Airport Schiphol (see Verbraeck et al. 1998, 1999, 2000). Here a number of groups worked at the same time on different components, such as the handling equipment and procedures, the vehicles, the conveyor belts, the infrastructure, the terminals, et cetera. During this project, all designs were compared by means of a simulation approach, and shown to other groups by means of visualisations. We found that the use of the visualised models fostered the discussion by providing a “living” representation instead of a written one.

Therefore, we have paid significant attention to the way processes are visualised in the model system. We propose the following guidelines concerning visualisation of processes:

− Apply a true-to-scale visualisation of physical processes.
− Create iconic representations of equipment states, such as “waiting for sequence” or “waiting for order”.
− Create a link between visualised process control components and physical processes by using the same symbols, colours, and scale.
− Create the ability to zoom in into specific processes in order to be able to explain the functionality and consequences of a design choice.
− Use, whenever, possible 3D visualisation to validate the physical movements.
− Do not focus too much on 3D visualisation, because it might distract people from the core of a certain design.

5.6 Verification and validation

When a design approach is heavily leaning on the use of models to support decisions, it is of utmost importance that the models that are used are without errors and are valid. Therefore, presenting a valid representation of reality (i.e., fidelity) is one of the basic requirements a model suite should meet. The project management should
request a proof of validity from the group applying the simulation approach before any decisions are made.

Kleijnen (1999) defines validation as “determining whether a simulation model is an acceptable representation of the real system – given the purpose of the simulation approach”. This means that a model can be valid for one purpose, but invalid for another. For creating a valid model – not to mention the steps that have to be taken to arrive at a correct model in itself, the actual verification – there are basically two different methods: expert validation and statistical validation (Kleijnen, 1999). It is recommendable to apply both in the context of automated container terminals. For further details concerning ways of validation we refer to Sol and Verbraeck (1992).
6 **Way of Controlling when Designing Robotized Container Terminals**

In this chapter, we pay attention to the interactions between the people involved in the design-engineering process. The orientation is process-oriented. We analyse which actors can be discerned regarding the application of a simulation environment as part of a design-engineering process of a container terminal. We discuss the relationships between the actors and their responsibilities within the design-engineering process.

6.1 **A concurrent simulation group**

In chapter 2, we already stated that there are many different parties and people involved in the design-engineering process of a complex system, such as a robotized container terminal. Because we adhere to an integrated design approach, a major task is to integrate the contribution of the various disciplines involved or at least to guarantee a constructive interaction between the various disciplines. For general process and project management approaches, we refer to De Bruijn and Ten Heuvelhof (2000), Pahl and Beitz (1995), and Hall (1960). However, we recommend organising a separate, concurrent group of people that focus on the support of the design-engineering process applying a simulation approach. This group, dedicated to the application of simulation, should be active throughout the entire design-engineering process as a concurrent activity to the actual design-engineering process.

The group should be responsible for verifying, validating, and assessing the various design alternatives in the on-going design-engineering process. As we have discussed in chapter 4, these activities are supported by means of a simulation approach. This means that this group has to develop a simulation environment that is able to test alternatives, prototypes algorithms, and so forth (Van der Heijden, 2002; see also section 2.7, the ECT case, and see section 7.7.2). The following aspects are considered with regard to such a separate team that performs the activities just mentioned:

- First, the group that performs the simulation approach is not part of any other design team and therefore neutral: it solely fulfils a supportive role in the design-engineering process. Neutrality is important, because usually there are many contradictions (and therefore, also tensions) between the various design tasks (and teams). There is a limited budget that has to be shared, there are limited
resources that also have to be shared, and each component should contribute more or less to the overall objectives, i.e. creating a terminal that meets the service requirements. The model system could as such fulfil an impartial assessment role to compare the quality of solutions or alternatives, without preference for a certain discipline.

- Secondly, the simulation group stands between all other groups and has to gather knowledge from the other groups in order to be able to create the models. This process fosters the information exchange process (1) in an earlier stage, because usually the interface discussions between system components take place later and (2) between all disciplines, because when discussing the models with the responsible group, issues of other groups, and dealing with other design tasks are brought up as well.

- Thirdly, a separate group can stay focussed on functionality and performance, whereas the focus of the design and implementation teams often tends to shift during the process from ‘getting the things to work according to the specifications’ to ‘getting the things to work’. Because the simulation group is not directly involved in the implementation, it can remain focussed on the functionality rather than on the solving of implementation problems.

- Finally, the simulation group is an intermediary between the various groups involved in the design-engineering process. Because the simulation environment contains an animation as well, it may serve as a way of convincing others that some things will work and others will not. In such cases, a visualisation is easier to discuss than a piece of paper on which the same topic is described.

6.2 Integrating the simulation approach into the design process

6.2.1 Modelling as indispensable activity within the design process

The major pillar of the design-engineering approach proposed in this research, is the application of a simulation approach – i.e. the application of the problem solving approach - throughout the entire process. Because models play such an important role in this approach, we pay additional attention to the way these models are developed and applied throughout the process. In the next sections, a series of recipes, rules,
and guidelines are discussed, which will be of use when applying a simulation approach to the design-engineering of a container terminal.

On top of the decision-making support for which the models are primarily used, Davenport (1993) mentions the following other goals of a simulation approach:

− "It should not only provide a descriptive, but also an analytical, model of the process, facilitating and understanding of such factors as time, cost, and other resources consumed by the process.

− It should enable the translation process of a business process model defined at a strategic level towards operational processes, as well as the definition of an incremental change process from the current situation towards the new – at strategic level envisioned – situation.

− It should support the addition of successive levels of systems and data-oriented details, enabling it ultimately to serve a useful purpose during the systems design and/or prototyping phases."

When these goals are taken into consideration, one can see that a simulation approach serves as a means to support decision making by providing information in such a way that it is understandable for the people who make the decisions. In order to define the requirements related to the interaction between the decision-maker, the model, and the modeller, we pay attention to the role-play during the development of these model systems, or more generally said “inquiry systems”.

### 6.2.2 Actor interaction in system development

When designing a simulation environment, three types of actors can be recognised according to Churchman (1971, p.43): the designer, the client, and the decision-maker. This statement could be generalised for all kinds of systems development, including the type of application of simulation that is proposed in this thesis. However, the life cycle of a system goes beyond the design and implementation. Therefore, another actor could be added (as is recognised by Bennetts et al., 2000): the operator and/or the administrator of a system, who comes into play after a system has been taken into operation.

---

22 Churchman consequently mentions an inquiry system where we prefer simulation environment or model system. Although an inquiry system is a more comprehensive notion, we focus in this research on the specific use of simulation models alone.
The four actors are archetypes or idealized role models whose actions are described by Churchman (1971) through the following statements (Bennetts et al., 2000):

− Each archetype can be one person or many people or all four can be found in one person (Churchman, 1971, p.48).
− Each archetype is assumed to have a recognised set of interest values, which are to be addressed by a simulation approach. This should be linked as well to the background of people involved, whether it is equipment design, software design, or an operational or managerial background.

An additional statement could be added from the perspective of designing a system that provides design support throughout the engineering process:

− The archetype's set of interest values can and will change during the design-engineering process due to a change in issues, change in levels of detail, and change of the people who fulfil the role. In a number of cases, we have experienced this behaviour in terms of different priorities, varying senses of urgency, different solutions, et cetera.

Regarding the role play around the design of a simulation environment, Churchman (1971, p.43) also defines a number of conditions that accomplish that the simulation environment can be conceived as a system:

− “The simulation environment is teleological” (i.e., the system serves certain goals, identifiable via a cause-effect relationship). When analysing the goals of the simulation environment proposed in this thesis, we can identify a number of goals, differing across the activities within the design-engineering process.
− “The simulation environment has a measure of performance” (i.e., there are some performance indicators which make it possible to evaluate the goal satisfaction or cause-effect relationship). This condition is an important one when taking into account the added value a simulation environment should provide regarding the decision making process specifically, and the design-engineering process in general.
− “There exists a client whose interests (values) are served by the simulation environment in such a manner that the higher the measure of performance, the

---

23 In this section all statements between parentheses are from Churchmann (1971) and reflected by Bennetts et al., 2000.
better the interests are served, and more generally, the client is the standard of performance”. As will be explained later, a simulation environment that proves to be valid will not per se fulfil the performance requirements of the client. Accreditation of a simulation model is almost as important as model validity.

− “The simulation environment has teleological components which co-produce the measure of performance of the system as a whole”. This statement is interesting because Churchman assumes that (1) the simulation environment can be divided into subsystems or components and that (2) each component’s contribution to the system’s performance exists or might even be identifiable.

− “The simulation environment has an external environment – which also co-produces the measure of performance of the system”; an almost obvious but still important statement: The extent to which a simulation environment is successful – measured by either objective performance measures or subjective assessments of the client or the decision-maker – certainly depends on what is happening during the development process. Varying priorities, changing budgets, competing systems can all be reasons for a lesser performance of the system.

− “There exists a decision-maker who – via his resources – can produce changes in the measures of performance of the system’s components and hence changes in the measure of the system’s performance”.

− “There exists a designer, who conceptualises the nature of the simulation environment in such a manner that the designer’s concepts potentially produce actions in the decision-maker’s interest, and hence changes in the measures of performance of the simulation environment’s components, and hence changes in the measure of performance of the simulation environment”. This statement is strongly related to the scope and purpose of the simulation approach. In order to provide the right answers the simulation environment should only cover those topics that are relevant for assessing various alternatives. We address this as the scope of a simulation approach. The designer of the simulation environment should be able to create the model system that is needed to meet this requirement.

− “The designer’s intention is to design or change the simulation environment so as to maximise the system’s value to the client”. Theoretically, this is true, but
in practice, it can be difficult to control system designers. Therefore, regular tuning between the client and the designer is strictly necessary to accomplish a simulation environment that meets the client’s requirements.

− “There is a built-in guarantee that the designer’s intentions are feasible”. The task should be feasible, otherwise constructing a simulation environment is useless and time-consuming. Therefore, before the development of an inquiry system starts, a feasibility check should be performed. This feasibility should always be seen in the scope of the analysis; building a model that takes two years, whereas the realisation would only take a few months, does in most cases not make sense.

Churchman gives the designer the central role within the model system development process by defining the conditions from a designer’s perspective. Although this is in line with the approach we proposed in chapter 4, where the design-engineering process is placed as the leading process when developing a container terminal, it requires a remark concerning the development of the simulation environment. In the setting as we propose here, the development and application of the simulation environment is partly a derivation of the real system, i.e. the equipment and its process control system. This will limit the freedom of the designers of the simulation environment, but even puts more emphasis on the guidelines from Churchmann, that are given below:

− “The designer has to identify the client and the decision-maker”. Depending on the coherence within the designer role as well as the coherence within the client and decision-maker roles – all roles can exist of more than one person, see definition of the archetypes – the identification of the client and the decision-maker can be very difficult, which is one of the main problems when designing a system (i.e. the container terminal).

− “The designer needs to have a theory about his role as well as a theory about the system. The designer must learn about the system and understand the influence he can and should have on the system that is required” (Churchman, 1971, p.52). In other words, a designer cannot properly design a system he is not familiar with in the sense that he does not understand how the system he is designing or building works and functions.
Furthermore, Churchman defines a number of conditions regarding the relationship between the designer and the client (Churchman, 1971, p.47-48):

- “The designer has a value structure identical to that of the client”. This condition aims at accomplishing that the designer considers the same topics important as the client. This condition hardly seems feasible as their backgrounds and role are different, but the client can contribute to this condition by clearly defining his requirements, so that the designer is then able to create an identical value structure.

- “As an actor in this scenario, the client is described only in terms of his value structure”. Because the client does not act in an active way during the development process other than by making his value structure explicit to the designer, his value structure is of utmost value. However, in practice, the interaction between designer and client is much livelier. This is necessary, because by the evolving design, the client might change his value structure.

- “The designer invokes a world in which the client could change whatever was wished within the bounds of limited sources”. A simulation environment is very likely to be used in different settings, for instance, it could be applied in the design of two different container terminals, of course sharing the major concepts, but certainly deviating at detailed level.

- “The designer seeks to describe the underlying principles of the client’s choices, using a measure of performance”. The simulation environment’s added value should be measurable in terms of the measures of performance – or performance indicators – of the client, not of the designer. Those measures as well as the relative weight of each measure also can and are likely to deviate between the designer, the client, and decision-maker.

- “The designer is successful to the extent that he can accurately measure the client’s real preferences”. Although an important condition for success, in practice this is one of the hardest conditions to satisfy, because the client’s preferences are often ambiguous and internally contradictory.

- “The designer must analyse possible futures by designing and, in principle, implementing each of them. The measure of performance is used to assign numerical values of these possible futures and hence rank them against each other”.

Chapter 6: Way of Controlling
“If the decision-maker’s ideas about a system are not seen as ‘good’ by the client, then the designer’s role is to try to change the decision-maker’s value structure”.

Finally, Churchman mentions a number of conditions concerning the relationship between the designer and the decision-maker (1971, p.48 and p.52):

- “The designer is expected to choose the decision-maker in a way that will maximise the measure of performance”. It can clearly be seen here that Churchman gives the designer a central role, although in practice the designer is very dependent of both the client and the decision-maker and it is unlikely that the designer will determine who will play the decision-maker’s role.

- “The designer’s ideas about a system are expected to produce changes in the actions of the decision-maker and hence changes in the measure of performance”. In the event that the balance of power between the three roles involved is equal, it is plausible that the designer’s ideas will have significant influence, but in most cases the client plays the dominant role when it comes to functionality and the decision-maker does with regard to time and money spent on the development.

- “The decision-maker co-produces the future along with the environment, which he does not control”. Although a decision-maker may clarify his interests and ideas, he still depends on the designer to transfer these ideas into the model system (simulation environment). This sometimes creates a lack of trust in the outcome of the analysis, especially when the benefits of a certain improvement measure appear to be lower than the decision-maker expected.

- “The environment is defined by what is not changed by the decision-maker. Decisions about what will be changed by a decision-maker and what will not be changed depend on the decision-maker”. This statement is questionable; a better statement would be that the environment (of the model system) is either defined by what cannot be changed by the decision-maker or by what is beyond the scope of the design process but still influences it. Therefore, the main influence on what can be changed and what cannot be changed is defined by the scope of the simulation approach. The decision-maker plays an important role in determining this scope.
“The decision-maker also has a value structure, but it is not necessarily the same as that of the other two (three including the operator/administrator) actors”. The value structure of the designer should be irrelevant, but in practice, the client and decision-maker should be aware of the designer’s interests as well.

6.2.3 The designer

As said, Churchman addresses the designer as key role in the systems development process, since he is the one who is the executor. Hirschheim and Klein (1989) have identified a number of metaphors classifying the role of the designer. Those metaphors each suit a different design process philosophy.

In the first philosophy, the designer (the same as mentioned in the previous sections) is seen as a systems expert. According to Hirschheim and Klein (1989), many systems have been developed successfully in this manner. In this philosophy the system design process is approached as a technical issue, with a clearly defined reality – consisting of objects, the object’s properties, and processes –, without ambiguous goals, multiple perspectives on the goals and performance measure, et cetera. The key players are the manager or decision-maker, the developer and the user of the system to be developed (Bennetts et al., 2000). This philosophy clearly has its shortcomings and that is why other philosophies also provide relevant ideas and perspectives on a systems developing process.

The second philosophy Hirschheim and Klein (1989) mention is almost the opposite of the first: the designer is seen as a process manager or facilitator. It is clear that departing from the assumption that every system development process takes place in an environment in which the goals are clear and homogenously shared by all the people involved, is unrealistic, although in our opinion it is the best environment in which a system can be designed. It is, therefore, the question whether and how the two approaches mentioned, can be combined in a way that organisational reality – to summarise the deviating goals, the different perspectives of value of the people involved, the strategic behaviour during the development process, et cetera – is taken into account.

Philosophy three tries to combine the two possible roles of the designer by defining the goals as improving managerial control (Bennetts et al., 2000). Such a goal is
principally a process-oriented goal in contradiction to the type of goals Churchman (1971) mentions. Furthermore, the fact that process goals are the principal goals implicitly means (1) that the goals can change during the process, (2) that they do not represent the value structure of anyone (including the managers themselves) and (3) that there certainly is not a teleological relationship with a rational (shared) goal definition that can be accomplished by developing the system. The role of the designer cannot be clearly defined in this philosophy.

6.3 Concluding remarks
The management of a complex project like the design and engineering of a container terminal is a project in itself. The use of a simulation approach throughout this process can support the decisions that have to be made by creating more insight into the way it will work (visualisation) as well as by quantifying benefits and costs. When a model system is designed, taking into account the guidelines based on Churchmann (1971) and Bennetts (2000), mentioned in this chapter, the chance that the simulation approach is effective, is likely to increase.

We recommend installing a separate group that focuses on the application of the simulation approach. The expected benefit is the ability to remain focussed on the accomplishment of the service requirements next to getting the operation going. Besides, the group has the responsibility to verify whether the designed functionality is transferred into the operational system.
7 APPLICATION: THE REDESIGN OF A PROCESS CONTROL SYSTEM

This chapter is the first chapter containing experiences and observations when applying our proposed approach in a real-life case. The case revisits ECT (also presented in section 2.7) and discusses differences and similarities between the approach back then, and in the actual setting. In the final section we reflect on our proposed guidelines in particular.

7.1 Application of design approach in real-life case

The case described in this chapter concerns the redesign and replacement project of the process control software (PCS) of the Delta terminals of ECT in Rotterdam. In section 2.7, the original design and development project of the Delta terminals has been described already. This process took place from 1987 until 1993. Although we will see in this chapter that the two design processes are different in many aspects (if only because of the fact that it concerns a software replacement project rather than a complete new terminal), it is interesting to use the former design process for comparison. Because the commissioning is planned to take place in 2005, we cannot compare the outcomes of both projects, and therefore, we will focus on the analysis of the system that is being replaced, the functional specification of the new process control system and the technical design of it.

The outline of the chapter is as follows. First, we define the scope and objectives of the replacement project and the followed approach. We also discuss our involvement in the process. As next step, we describe the conceptualisation of the model system. Subsequently, we discuss the verification and validation, followed by the experimentation and model treatment. We end this chapter with the findings concerning the application of our proposed approach during the test case.

7.2 Scope and objectives of the PCS replacement project

The redesign and replacement project of the process control system (PCS) addresses a number of problems at the ECT Delta terminals (see Connekt, 2002).

First, there is the need to improve the berth productivity in terms of absolute performance but especially in terms of performance consistency. In the actual situation, the berth productivity is regularly meeting the requirements, but sometimes the berth productivity remains below the requirements. This causes delays in a rather
unpredictable way, which does not contribute to the reliability of the service towards the shipping lines.

Figure 7-1: Overview of the Delta peninsula (source: ECT)

The second issue that is addressed by the PCS redesign and replacement project is the objective of ECT to integrate the operations of the three Delta terminals. The idea behind this integration is that it will be easier to move vessels from one terminal to another and to share the yard capacity between the terminals. This will lead to the possibility to increase the berth occupancy without increasing the vessel service times (see also section 4.2.3), and therefore, create the opportunity to increase the terminal throughput capacity. In the current situation, there are three robotized ECT terminals in operation at the Maasvlakte, but two of them (Delta Dedicated East – DDE – and Delta Dedicated West – DDW) are not physically separated (see Figure 7-1). However, the current PCS does not allow real-time exchange of equipment between the two Southern terminals. The Delta Dedicated North (DDN) is situated on the other quay of the peninsula, and therefore geographically not adjacent to the other two. However, an integrated operation – including the possibility to move vessels from the north side to the south side - of all three terminals, especially when berthing is concerned, is thought to be beneficial for ECT and its customers, because
the flexibility is increased. The current software does not provide these possibilities and it would require a radical redesign to allow for such an operation.

These two problems, and a number of smaller issues that are not mentioned because of being irrelevant for this research, have brought the management of ECT to the decision to replace the PCS software.

7.3 Design approach entire project

7.3.1 Process activity plan

The design approach chosen in the software replacement project is a straightforward approach; similar to what is often seen in these kinds of projects (see e.g., Pahl and Beitz, 1995). The following steps are taken:

- Analysis of the current processes and currently operational software (section 7.4 and 7.5).
- Functional specification of new process control software (PCS).
- Technical design of the new PCS.
- Software development.
- Testing, commissioning of the new PCS, and training of users.
- Start of operation with new PCS (all section 7.6).

The estimated project duration is 2-3 years, which is relatively short for such far-reaching changes. The software that will be applied is for the major part standard software that will be (re)configured for this terminal. However, in comparison with most other terminals, the processes at ECT are different, because of the robotized concept with RMGs – addressed as ASCs in ECT terms - and AGVs and the connection via a manually operated straddle carrier system to rail and road. Most probably, this will require more changes of the standardised software than usual for a terminal of this size.

All the project activities are carried out by a consortium of software vendors that delivers the new software to ECT. ECT supplies information about the current

---

24 Furthermore, simulations have shown that it is better (in terms of vessel service capacity) to be able to distribute vessels over the various quays, than to be obliged to berth vessels fixed on a certain quay (see Dohner et al, 2002b).

25 This consortium consists of Gottwald Port Technology GmbH and Navis LLC.
system, especially about the hardware and its on-board software that will remain in place.

7.3.2 Application of a simulation approach
In this project, an explicit task and delivery is defined for analysing the current and future processes by means of a simulation approach. A separate group is responsible for this approach. The following steps that are taken to support each design activity are to be mentioned:

− Creation of a conceptual model.
− Transforming the conceptual model into a model system, containing all objects and data relevant for the pending design-engineering process (see Figure 7-3).
− Verification and validation of the model system.
− With the model system, various types of analyses – also addressed as experiments (see e.g. Meel, 1994) - are carried out, determined by the type of design activity. Based on the outcome of these experiments, design decisions can be made or supported.

This process is repeated until the design meets the requirements under a series of operational scenarios that are considered representative for the various operations at the Delta terminals.

7.3.3 Role of the researcher in the design process
The research approach applied in the test case can be classified as action research (see also section 2.9.3), since we participate in the PCS redesign project and influence the design approach and the results. However, the benefit of applying a simulation approach in the way described above is assessed by many others than the researchers themselves (see also the results of the expert survey in chapter 9). The assessment of the new PCS takes place in terms of objective performance indicators.

7.4 Analysis of the “old PCS”

7.4.1 Approach
The first step in the PCS redesign and replacement project is to determine the bottlenecks in the actual system, and subsequently define the priorities for improvement. Based on experience of the previous 10 years of operation, many ideas for improvement exist within the ECT organisation. The question is to determine
which improvements should be given priority, based on their potential contribution to the terminal (berth) productivity increase.

The analysis of the current situation and system – both the hardware and software – is carried out by means of the following activities:

− Interviews with operational and system experts.
− Study of the documentation of the control system.
− Analysis of the way the performance can be measured.
− Study of the specifications of equipment.

Figure 7-2: Aerial view of the one (DDE) of the Delta terminals of ECT (Source: ECT)

The challenge in this project is to model the current PCS in such a way that the model will be a valid basis for testing improvement measures. The available documentation about the PCS is only limitedly valid and not complete, because during the last ten years the control system has been continuously changed on minor not well documented points. Moreover, the knowledge about these changes is scattered throughout the organisation, if not absent. Therefore, the validation is done in co-operation with operation experts from ECT. Besides, a comparison is made.
with an emulation of the real system\textsuperscript{26}, which is in essence the real software, running with simulated equipment.

7.4.2 Creating a model system

First, the processes that take place at the Delta terminal are modelled in the simulation environment (see for an aggregated overview of the main process flow Figure 7-3). In this modelling process, a number of reductions are made. Based on discussions with operational experts from ECT and after considering the results of the validation experiments with the model, the reductions and assumptions are considered to be valid for the purpose of the analysis. Below, the main assumptions and reductions are discussed.

The model system is limited to the processes in between the quay cranes (QC) and the landside transfer points of the stacking crane (ASC). This means that the following physical processes are represented in the model system:

- The quay crane (QC) process.
- The interchange process between QC and AGV or vice versa.
- The driving of the AGV.
- The interchange process between ASC and AGV or vice versa.
- The ASC process.
- The interchange process between SC and landside transfer point.
- The arrival process of trucks at their transfer points.

Not taking the landside handling system (by means of straddle carriers – see Figure 7-4) into account, means that nothing can be said concerning the assignment of straddle carriers, and therefore, the exact handling times of trucks at the transfer points. However, since the straddle carriers are not considered to be the limiting factor for the truck handling time, the duration of the ASC move to deliver a container to the transfer point or to pick-up a container from the transfer point, is considered to be representative for the truck handling time.

\textsuperscript{26} ECT calls this test environment the Performance Evaluation Programme (PEP), which is a copy of the real software, combined with simulators of the equipment (QC, AGV, and ASC).
Besides, not all container flows are considered. Only the containers from and to the main stack (the automated yard with ASCs) are considered, leaving away the (small) flow to the multi-purpose stack at the DDN.

Hatch cover handling, handling of special cargo and restows are not considered either. Although they influence the overall productivity, there is no direct effect on the operation as such. During these operations, the QC does not require container supply or pick-up, which allows for assigning equipment to other QCs.
In order to create an identical experimental set-up the same data for the stochastic processes is used in the emulation and the simulation. The data comprises:

- The initial stack configuration (all containers, their location, and weight); usually the stack is generated in a random way, distributing the containers over the yard following an empirical pattern.

- The load lists and discharge lists; usually the containers are chosen from the yard according to certain rules, for instance exchanging similar (same voyage, same port of destination, same weight class) containers to obtain a random distribution over the various stack modules in time.

- The landside demand distribution; here a negative exponential inter arrival time is used with an inter arrival time (in seconds) equal to \(3600 / \text{number of trucks in a specific hour}\).

- The quay crane cycle times; usually they are created from an empirical distribution that is based on measurements from representative quay cranes (see also Figure 7-5). For validation reasons, they also have been based on the (fixed) cycle times that are used in the QC within the emulation.
The outcome of the emulation and the simulation is compared in terms of quay crane productivity. The comparison focuses on an eight quay crane operation, four loading cranes and four discharging cranes. This operation is chosen, because a mixed operation, i.e. loading and discharge at the same time, is the most representative operation. An operation with eight quay cranes at one terminal seldom occurs, but is chosen to put a heavy load on the system and to see where the bottlenecks show up.

The average quay crane cycle time is 77 seconds, corresponding with an average productivity of 46.7 moves per hour (mph). The cycle time distributions of the quay cranes in the simulation and the emulation are compared; the result is shown in Figure 7-5: the average is identical; the distribution slightly differs. An additional experiment, using the same cycle times as within the emulation, gave the same results, and therefore, the differences were considered to be insignificant.

When the outcome of the simulation was sufficiently close to the outcome of the emulation – in the opinion of the operational experts of ECT – the deviation

---

27 This is the quay crane productivity that would result as average over multiple hours, in the case that the QC does not have to wait for an AGV. Therefore, this is the highest outcome of the simulation experiment with regard to the QC productivity.
between the realised QC productivities was in the range of ± 5%, the bottlenecks within the current operation were analysed.

**Figure 7-6**: Effect of the number of truck arrivals at the terminal on the quay crane productivity; the quay cranes are able to do 35 mph each. In the actual peak at ECT (2001), the ASCs have to handle approximately 90 mph at the landside per terminal. With more than 120 truck arrivals per hour at the landside, the quay crane productivity is negatively influenced. With more than 240 truck arrivals per hour, the productivity of the ASC decreased, because all jobs get the highest priority, making the ASC travelling up and down from waterside to landside and back (Saanen and the Waal, 2001).

### 7.4.3 Model treatment

Although the interviews with the ECT experts already led to a long list of perceived bottlenecks, the analysis by means of simulation separates the real bottlenecks from the smaller problems. In this manner, a prioritisation of problems can be made. Some of the bottlenecks appear to be real bottlenecks, others, such as the influence of landside truck arrivals on the waterside productivity (see Figure 7-6), appear to be of lesser importance when improving the control system, because the perceived negative effect on the quay crane productivity is less than 2% and therefore considered to be insignificant.
Table 7-1: Overview of effects of evaluated improvement measures

<table>
<thead>
<tr>
<th>Improvement measure</th>
<th>Increase of service level to QC²⁸</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average performance (without disturbances) base situation</td>
<td>81.0%</td>
</tr>
<tr>
<td>Twin carry (2 x 20 ft on AGV), 10% of the moves</td>
<td>3.5%</td>
</tr>
<tr>
<td>Twin lift (2 x 20 ft by quay crane), 10% of the moves</td>
<td>8.8%</td>
</tr>
<tr>
<td>Increase of AGV speed to 4 m/s / (5 m/s)</td>
<td>5.2% / (8.8%)</td>
</tr>
<tr>
<td>ASC move scheduling</td>
<td>3.5%</td>
</tr>
<tr>
<td>Allowing swapping of containers at quay crane</td>
<td>7.0%</td>
</tr>
<tr>
<td>Reduction of variation in quay crane cycle time distribution</td>
<td>3.5%</td>
</tr>
<tr>
<td>Improved AGV job assignment</td>
<td>5.2%</td>
</tr>
<tr>
<td>Improved AGV turning behaviour (smaller claims)</td>
<td>3.5%</td>
</tr>
<tr>
<td>Early start-up message to AGV</td>
<td>7.0%</td>
</tr>
</tbody>
</table>

After the validation and the bottleneck analysis, the modelling and assessment of improvement measures started. Many improvement measures are analysed with

²⁸ The standard experiment consisted of 8 quay cranes, 32 stacking cranes (ASC) and 60 AGVs. 4 quay cranes were loading, 4 were discharging. A quay crane productivity increase of 1 mph (move per hour or box per hour) means that in average one quay crane has moved one container more per hour from or to the vessel. The maximum attainable performance in the simulation of the quay cranes was 35.2 mph.

²⁹ The measured base performance is the net productivity of the quay cranes and not the overall gross productivity of the terminal. This means that in practice all losses due to large disturbances, wrong information, meal-breaks and other stops, vessel movements, mooring, unmooring, are not considered in this productivity. Furthermore, in reality an operation with 8 quay cranes is a rare operation. This means that in practice the net productivity might be higher than the measured 28.5 mph. However, the total terminal production (quay cranes x productivity per quay crane) will not exceed 8 x 28.5 (228 mph) in reality. The peak values with the old PCS are around 160-165 berth moves per hour with 6 quay cranes (approx. 27 mph per crane).

³⁰ Only in combination with twin-carry. In effect the number of crane cycles per hour decreases (to approximately 28.0 mph), whereas the number of containers per hour increases (to approximately 31.0 containers per hour).

³¹ AGV base speed in straight line is 3 m/s; increasing the speed to 6 m/s does not improve productivity in comparison with 5 m/s. Speed in curves is 2 m/s in base situation and 3 m/s in all improvement scenarios.

³² In the current situation, the ASC started when the AGV receives its last claim before arriving at the transfer point. The improvement measure triggers the ASC in such a way that it is ready when the AGV is planned to arrive.

³³ In the current situation, the AGVs have to arrive in the right sequence when loading the vessel. Allowing the AGVs to exchange their arriving sequence with up to 4 successors or predecessors, leads to the presented productivity increase.

³⁴ The QC cycle time distribution ranges from 40 to 350 seconds, which makes the scheduling of AGV arrivals complicated to such an extent that always a buffer of AGVs is needed to avoid the QC to wait.

³⁵ In the current situation, the AGV job assignment algorithm takes primarily the distance from the actual AGV location to the pick-up location of the container into account. However, due to the layout of the terminal, the distance of an entire move (empty plus loaded part) is always the same, which makes distance the wrong parameter to optimize on. The alternative assignment takes the urgency of the move, the number of AGVs underway, and the sequence into account.

³⁶ In the current situation the AGVs need a large space when they turn. The space occupies adjacent lanes and therefore reduces the throughput of the lanes. The turning behaviour could be altered in such a way that the curves occupy less space.

³⁷ In the current situation, the AGV starts-up after it has received its route or claim. However, after longer times of standstill, it takes up to one minute to get started. A ‘wake-up’ call in advance could reduce these time losses due to start-up.
regard to their effect on the quay crane productivity, which resulted in a priority list for the future process control system. This priority list was based on the relative contribution of improvement measures to the quay crane productivity, as well as on the feasibility (e.g., cost, duration) to implement the solution. In Table 7-1, an overview of the main improvement measures is given, including their contribution to the quay crane productivity when applied as single measure. In combination with other measures, the improvement for single measures is less, which is caused by the fact that the quay crane approaches its saturation level of 35 mph. An improvement measure not discussed here is the anti-congestion algorithm, a project that was being commissioned at the time. This alternative is discussed in more detail in section 7.5.

![Figure 7-7: Two packages of improvement measures. One package with changes that only affect the software; one package that also required adjustments to the hardware (equipment) and the embedded software running within the equipment.](image)

All improvement measures that appeared to contribute to the performance of the terminal, were specified by ECT in the functional requirements for the software redesign and replacement project. These functional requirements were handed over to the software suppliers.
7.5 Analysis of the congestion-free router in combination with “old PCS”

7.5.1 Approach

Besides the improvement measures mentioned in the previous section, there is one additional measure that was being implemented at the time of the bottleneck analysis. In the old PCS, the AGVs suffer from congestion, which causes badly predictable travel times. The unreliability of the AGV travel times causes waiting times for the adjacent equipment, i.e. stacking cranes (ASC) and quay cranes (QC), or as is the case at ECT the need for more AGVs to achieve the same quay crane productivity. A method to solve the unreliable travel times is congestion-free driving. Congestion-free driving means that an AGV trip is scheduled in space so that AGVs never have to wait during their route execution. The idea behind this concept is that AGVs do not start their route as soon as possible (as they did with the old PCS), but that they wait until a congestion-free route is available. However, the question remains whether the application of this concept increases the performance of the entire process (measured by the quay crane productivity), since the concept might have side effects, such as a higher occupation of transfer points at the stack (due to the waiting for a congestion-free route) and problems finding a congestion-free route for difficult origin-destination pairs resulting in long waiting time and thus a reduced AGV productivity. Therefore, the question is whether the application of congestion free driving is beneficial for the overall terminal performance. In order to answer this question, the same model system was used, with the exception that the router was exchanged with a congestion-free router. The model is described in the next section.

7.5.2 Creating a model system

A simulation approach is used to assess the impact of this algorithm on the quay crane productivity. The system environment of this route scheduling component was already available in the form of the model system of the current ECT terminal, including most planning and control algorithms (see section 7.4). Two additional algorithms had to be modelled: a route finder – the existing router in the old PCS was a static router using a table with all the available routes – and a route scheduler. Together these algorithms are addressed as the anti-congestion algorithm.
The task is to compare the anti-congestion algorithm with the current router (the router of the old PCS); a route scheduler did not exist in the old PCS, because the AGVs started as soon as they received the route. Moreover, the anti-congestion algorithm has to obey all present constraints. The most important constraint is the loading sequence of the containers according to the vessel stowage plan. Because the layout at ECT does not allow for overtaking at the lanes under the quay cranes (see Figure 7-8), the AGVs have to arrive in the right sequence at these lanes. In the current situation, this is solved in a very simple way: an AGV waits until its predecessor has passed the point – called the check point (the entry lanes in Figure 7-8) – where their routes join. In the anti-congestion algorithm, the route scheduling solves this by assigning the start time of a route such that enough time remains between the arrival times of AGVs at the quay crane considering the driving time of each route.
In formula:

$$\text{TimeOfDeparture}_{i+1} + \text{EstDriveTime}_{i+1} \geq \text{TimeOfDeparture}_i + \text{EstDriveTime}_i + \delta t$$

The assumption behind the route scheduling is that the AGVs are capable of executing the route with certain reliability in terms of duration and occurrence of disturbances. The possible difference between actual driving time and estimated driving time is solved by adding a margin $\delta t$ (2 seconds) to cope with small deviations due to limited equipment accuracy.

7.5.3 Model treatment

Without going into detail concerning the actual implementation of the congestion-free driving algorithm in the simulation (see for further information Verbraeck and Saanen, 2004), we discuss the effects of the solution. Similar to the validation of the model of the old PCS, the new component was validated by comparing the model system with the emulation of the old PCS + the congestion-free driving algorithm. After the model proved to be valid for the purpose of comparison, the experimentation started.

The experiments are carried out with the same layout as is used in the actual situation, i.e. a limited number of access lanes to and from the quay cranes. However, the driving direction is changed according to the typical pattern that the new solution allows. The entire solution that was implemented at ECT allows for more flexibility in creating access lanes, but in order to assess the benefit of the anti-congestion algorithm, we decided to test it in a situation with a similar layout.

In addition to the effect on quay crane productivity, it is necessary to analyse the effect on behaviour of the AGVs. Therefore, an analysis was made of the AGV-state during the operation. As can be seen in Figure 7-9, the AGV state distribution changes heavily due to the application of a congestion-free routing algorithm. First, we see that the AGVs drive less (from 47% to 25%). However, the average speed increases (almost times two), because the AGVs do not interfere with each other anymore (see “Waiting for Claim”). The decrease in driving is transformed in waiting

---

38 The actual solution—addressed as Dynacore—comprised more than just the anti-congestion algorithm. The other improvements were also implemented in the simulation environment and were validated by means of comparison with the emulation of the Dynacore system.
time ("Waiting ASC buffer” and “Waiting for Route”). The small productivity increase in this scenario (see also Figure 7-10) can be read from the slight increase of the “Transhipment time”, which is proportional to the number of moves at the quay crane. The same holds for the time in the QC buffer.

![AGV status diagram](image)

**Figure 7-9: Frequency distribution of AGV states with the old PCS and with the old PCS + the congestion free driving of AGVs.**

In Figure 7-10, the effect on the quay crane productivity can be seen. The maximum productivity that can be achieved is 35 mph. The relative performance (“service level”) to the maximum is also shown. In a mixed scenario (combined loading and discharge), there is an advantage of one move per hour per QC in the congestion-free situation over the old situation. However, this is mainly due to the improved performance of the discharge quay cranes. This is confirmed by the results from the two dedicated scenarios: in the event that all quay cranes are loading, the productivity decreases with more than one move per hour. This is because congestion-free routing scheduling is increasingly difficult when the routes are longer. During loading, the routes originate from the entire stack (one kilometre width), whereas during discharge a stack module close to the quay crane is selected. In the case of discharge – with the shorter routes; the difference between loading and discharge is on average 200 m (20%) – the benefit of the anti-congestion algorithm is almost 2 moves per hour per QC, a productivity increase of 6%.
Figure 7-10: Comparison of PCS old router and congestion-free routing in three different scenarios

The conclusion is with the layout used in the experiments - two operational areas with a limited number of access lanes - the congestion-free route scheduling does not significantly contribute to a performance improvement.

In addition to the assessment of the algorithm, the simulation approach showed that with a relatively small effort - approximately two weeks of algorithm development and two weeks of testing and experimentation - a complicated component can be prototyped and evaluated.

7.6 Design of the new PCS

7.6.1 Approach

The software engineering project is divided into four activities, which are executed partly at the same time, partly sequentially (see Figure 7-11):

1. Functional specification, consisting of the software architecture, the interface design, and a description of the functionality of the software system’s components.
2. Technical design of software components.
3. Implementation.
4. Testing, integration testing, commissioning testing.
After the commissioning tests, the system should go in operation using a “big bang” scenario: a switch between the old system and the new system within 24 hours.

The team responsible for testing the functional and technical design on its contribution to the performance requirements is a separate team, operating in parallel with the groups responsible for the design and implementation activities. There is a continuous interaction between the design and implementation teams and the team performing the simulations. From customer side (ECT), there is a group involved with the validation of the simulations and the definition of operational scenarios, relevant for testing the system (see Figure 7-11).

7.6.2 Conceptualisation

The software redesign and replacement project touches all terminal components; from the gate process to the waterside processes, from the operational control to the billing and data exchange with customers. Without a complete overview, where all interactions between the various components are depicted, a new software architecture can hardly be made. Therefore, much attention is paid to the documentation and (re)design of the business processes and the interaction with software components. This process has merely a top-down character, rather than bottom up. For the client (see also chapter 6), the container flows through the terminal are the way to check whether all business processes are covered by the new software design. Again, the container flows touch most terminal components and processes and therefore a holistic design perspective is required to create the overview and to be able to check whether all processes are covered in the new design.

Because all business processes that were supported in the old system have to be present in the new system, the blueprint of these business processes is taken as the basis for the new software. Therefore, the operational processes are definitively leading in this project, although they are not considered as untouchable; in the case that a significant productivity gain can be achieved with a change of the software and operational procedures, the operational procedures are reconsidered.

---

39 This is true for the team that develops the software for controlling the automated equipment. The planning software is developed without a separate group that first tests the design on performance.
Furthermore, the actual operational processes form the requirements for the new software, because many processes are supported by present hardware and cannot be easily changed without also altering the hardware, which is outside the scope of the project.

![Figure 7-11: Interaction software engineering approach, simulation approach and validation activities](image)

### 7.6.3 Creating a model system

The conceptual models of the new PCS software are all made in an object-oriented way, taking the various system parts as building blocks, e.g., the QC-system, the AGV-system, the ASC-system, the SC-system, the gate system, and the interchange
objects\textsuperscript{40} (see Figure 4-11, which shows at execution control level the structure of the new PCS). Furthermore, the software is merely written in C++. Because the conceptual models represent the structure of the physical subsystems and their interaction well, i.e. the real system and the information system have a similar architecture, the implementation process is eased, because the structure of the software can easily be understood, by looking at the operation and the layout. The explicit modelling of the interchange objects makes the software easier to re-use for other terminals, because the specific linkage between the subsystems leads only to specific software in the interchange objects.

\textbf{Figure 7-12: Integrated (upper figure) and split model system}

To be able to assess the quality of particular software components, the model system is similarly structured as the real software. However, the model system that is used during the analysis can only partly be re-used, because the new PCS software has a different architecture than the existing PCS. A major difference is caused by the fact that the new PCS software is delivered by two suppliers; one responsible for the overall planning and administrative functionality\textsuperscript{41} and one for the execution of jobs of the automated equipment\textsuperscript{42} (see Figure 7-12). This leads to a need for an interface between the two parts that are delivered by the two software suppliers. Because this

\textsuperscript{40}Here the objects are meant that handle the interchange of containers between two different types of equipment, i.e. the AGV and the QC, the AGV and the ASC, the SC and the ASC or the truck and the SC.

\textsuperscript{41}The supplier is Navis LLC.

\textsuperscript{42}The supplier is Gottwald Port Technology GmbH.
interface will not only have a major impact on the software architecture, but also on the technical and functional performance, the interface has to be implemented in the simulation environment as well. In order to make the separation between both software parts transparent, the interface is implemented similar to the real interface, i.e. via TCP/IP using the same messages, and the two parts of the control system are separated in the simulation environment as well (see Figure 7-12), creating a distributed model system.

In the model system, a number of elements of dynamic behaviour have been taken into account (see Table 7-2) that are not common in simulation models of container terminals (Bokkers, 2002; but see also e.g. Liu, 2001). However, they are considered of influence when assessing the quality of planning and control algorithms.

In order to model these dynamics, many data are gathered and analysed from the current operation. In the model system, some data are represented by stochastic distributions; other data were modelled by means of an empirical distribution, because the fit between mathematical distributions and the empirical data was not sufficient.

![Figure 7-13: On-line connection between simulated parts of the PCS and production parts under development.](image-url)
### Table 7-2: Comparison between modelling of specific behaviour in container terminal processes

<table>
<thead>
<tr>
<th>Actual situation</th>
<th>Common in simulations</th>
<th>Simulations in ECT project</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCs move in gantry direction.</td>
<td>QCs stationary.</td>
<td>QCs move according to bay plan.</td>
</tr>
<tr>
<td>Topology changes.</td>
<td>Topology frozen.</td>
<td>Topology changes according to QC positions and blocked areas.</td>
</tr>
<tr>
<td>‘Clean’ QC production of 20 - 30 moves per hour.</td>
<td>‘Clean’ QC production of 40 moves per hour.</td>
<td>Variable production, configurable per QC, based on empirical data.</td>
</tr>
<tr>
<td>Capabilities of crane drivers or QCs differ.</td>
<td>All QCs have same cycle time distribution.</td>
<td>Configurable per crane.</td>
</tr>
<tr>
<td>Occasionally long QC cycles.</td>
<td>Cycle time distribution with small variance.</td>
<td>According to practice (with occasionally long cycles).</td>
</tr>
<tr>
<td>Correlation in the cycle times of subsequent moves.</td>
<td>Independent samples from a cycle time distribution.</td>
<td>Correlation modelled, however discarded after statistical comparison.</td>
</tr>
<tr>
<td>QCs have interruptions, like meal breaks, hatch cover advancements, break bulk, hoisting/lowering the boom.</td>
<td>QCs handle containers continuously.</td>
<td>Continuous operation, however with possibility to handle hatch covers.</td>
</tr>
<tr>
<td>Equipment breaks down.</td>
<td>Equipment is perfect.</td>
<td>Equipment breaks down according to empirical data.</td>
</tr>
<tr>
<td>Communication disturbed or delayed.</td>
<td>Perfect communication.</td>
<td>Disturbance modelled.</td>
</tr>
<tr>
<td>Equipment is not exactly according to specification.</td>
<td>Exactly uniformly accelerated and decelerated movements.</td>
<td>Behaviour according to practice.</td>
</tr>
<tr>
<td>Equipment has start up time.</td>
<td>Equipment starts immediately.</td>
<td>Start-up times, depending on idle time (based on empirical data).</td>
</tr>
<tr>
<td>Operators need to prepare and hand over orders and are sometimes late.</td>
<td>Orders are always available.</td>
<td>Orders always available.</td>
</tr>
<tr>
<td>Some work needs to be cancelled.</td>
<td>Orders can always be executed as planned.</td>
<td>Cancellation possible.</td>
</tr>
<tr>
<td>The checker needs to enter some data and sometimes delays the process.</td>
<td>Checker is not modelled or acts always in time.</td>
<td>Variation in behaviour modelled.</td>
</tr>
<tr>
<td>Reefer handling including manual connecting and disconnecting.</td>
<td>Only standard containers.</td>
<td>Includes reefers assuming that connecting is dynamic.</td>
</tr>
<tr>
<td>Origin and destination of containers.</td>
<td>Uniformly distributed or random position selection.</td>
<td>According to real decision rules concerning yard assignment.</td>
</tr>
<tr>
<td>Operators making sub-optimal decision and/or late interventions</td>
<td>Well defined scenario with optimal topology, buffer strategy etc.</td>
<td>Sub-optimal decisions not considered.</td>
</tr>
</tbody>
</table>
7.6.4 *Experimentation, model treatment and outcome*

The model system was used for various purposes during the software engineering project. First, it was used to test whether the terminal is able to meet the performance requirements when controlled by the new system, as specified in the functional design. This was done by modelling the new components and algorithms and testing the whole on performance in various operational scenarios. The modelling required already a first technical design of each component (a prototype design) to depict the behaviour correctly. This means that after the functional testing, a prototype existed of almost all components that are relevant for performance testing. These prototypes were used as the basis for the technical design by transferring the design as pseudo-code to the implementation teams. Later adjustments, made during the software implementation, were fed back into the model system to keep it up-to-date with the actual software design. In a later stage, the model system was used to test software components on-line, i.e. linked with the model system. The first testing, for which it was used, was the fulfilment of a role as counterpart of the software that executes the planning. Because this software is based on a standard product, a first prototype was earlier ready than the custom-made software that controls the automated equipment (see Figure 7-12 and Figure 7-13).

Because the architecture of the model system was similar to the real production software, the interface between the planning components and the execution components were similar as well. This created the opportunity to link the model system to the real system in a later stage. Of course, functionality that was not implemented in the model system – mostly because it was considered not to be of importance to assess the performance of the new system - cannot be tested, even not in a situation in which the production software is linked to the model system. For both suppliers involved, the possibility to link their production software under development, provides a possibility to test in an earlier stage than possible without the availability of this kind of simulation environment (see Figure 7-13).

Although the simulated counterpart is a black box from a functional point of view, it contains much more realism than for instance a message simulator that only generates the messages of the counterpart. Although a message simulator is also useful – in this project a message simulator was created as well to support simple test scenarios –, it lacks the typical dynamic effects that occur in the real system. Take for
instance the reaction of the planning – implemented in one part of the software – on a delay message of the execution control – implemented in the counterpart of the software. It could be that the planning algorithm sees that the equipment is not capable of arriving on time. Therefore, it decides to assign another piece of equipment to this particular job. This change leads to a number of job re-assignments, which are sent to the execution control. Because the execution system had tried to optimize the execution of routes and the routes are already executed, now a number of new delays are introduced because vehicles have received new jobs and therefore new routes, which coincidentally require turning, which causes hinder for other vehicles. This results in new delay notifications to the planning, which leads again to a re-planning, a re-assignment, and so forth.

Next to testing and functioning as counterpart for the production software, the model system was used as tool for evaluating control algorithms and decision rules. This use was both internally – to support the assessment of various design possibilities – and externally – to inform the client and create confidence in the progress and result of the project. The following alternatives were compared:

- Effect of AGV speed; the fleet at ECT consists of AGVs that can drive 3 m/s, and AGVs that can drive 4 m/s. The question is whether a mixed fleet performs better than a fleet driving at only 3 m/s.

- Effect of the configuration of the layout (also called topology). Since there is limited space on the apron, and all AGVs have to enter and exit the area in between the crane legs, the capacity of the area is critical. Furthermore, there are multiple configurations possible, even for a single situation.

- Effect of the routing algorithm. There is space for up to 6 highways (see Figure 7-4), but it is questionable whether this is the best solution, since in this case the AGVs will drive closer to each other than with 4 highways, because the available space for the highways is 24 metres. When driving straight, this is not an issue, but during the curves entering or leaving the highways, they disturb the adjacent lane if that is closer than 6 meters.

- Effect of various container sequence control algorithms. It may be possible to change the arriving sequence of containers, when the stowage plan can be corrected dynamically. This allows AGVs that are early with the “wrong”
container to drive to the quay crane before late AGVs with the “right” container arrive.

Based on the outcome of these experiments, the final algorithms were designed and implemented.

7.7 Findings and conclusions

The PCS redesign and replacement project provided a good opportunity for applying a number of the guidelines we have proposed as part of the design approach. Although it is a project where we participated in the role of action researcher, it gave us the possibility to act according to the guidelines and to find out to which degree they are applicable. In the following sections, we summarise the experiences drawn from this case.

7.7.1 Using a simulation approach throughout the design-engineering process

The application of a simulation approach during the various activities throughout the analysis, design and implementation of the process control system at ECT, using the same model system as a basis throughout the process, resulted in the following findings:

- A reduction of development time, and verification and validation effort, because the same components are used multiple times.
- More consistency in the representation, method of data analysis and so forth. Although this can be achieved when different models are used, it is much easier when the models are re-used.
- Faster availability of the entire system environment, because the starting point is already a configured terminal with the all the equipment in place.

The last finding is important when the same model is used throughout the entire process, because then, the other activities will not be delayed because first a test environment has to be developed. Since it is not common to use models during the implementation of software, the advantages might even be a prerequisite for application, because there is often no time to wait for extensive model development when the software is about to be implemented. The experienced benefits during the implementation are the following:

- Visualisation of algorithms, which eases understanding and communication, especially for operators and management.
− Possibility to evaluate software both on its contribution to the terminal’s performance and technical performance.
− Possibility to find out whether certain solutions work (again both functionally and technically) much faster than if real software has to be developed.
− The availability of the entire system environment, fed with real data such as truck arrivals, vessel arrivals and so on.
− The possibility to freely configure the system in order to represent a certain situation, which has to be tested.

In general, the project showed that the use of the same model system is appropriate and beneficial during various activities of a major software replacement project. The project also shows that a simulation approach can serve various purposes, from answering what-if questions; to the testing of software, and prototyping. A simulation environment, which mirrors the real system architecture, can contribute to the reduction of development time and improve the quality of the (software) design.

7.7.2 Concurrent simulation group

The somewhat distant position of a concurrent group gives the possibility to focus on the task, which is to find a solution that is able to meet the functional requirements, without being disturbed by daily concerns about the implementation difficulties. As long as the interaction between the simulation group and the design and implementation team(s) is based on trust and mutual sharing of information, the approach with a concurrent group appears to be beneficial.

A problem that we experienced is the interaction between the developments taking place in the design process of the real system and at the same time in the development process of the model system. A bi-directional feedback loop is required to be able to test the right design and to provide insight in the consequences for performance, which inevitably leads to a delay. This delay can be kept as short as possible by regular communication and by a way of modelling that is close to the actual system architecture. The latter appears valid because the system architecture does not change too much after a certain point in time – preferably as early as possible in the project – which reduces the effect of changes of one system component on other components. When the software architecture and model system architecture are similar, the duration of the feedback loop can be reduced.
7.7.3 Architecture of model system and real system

We found that both during the analysis of the current system and the design of the new software, the identical architecture of the model system compared to the existing and new system, contributed to the efficiency of the design process.

- First, it eased the verification and validation during the analysis of the currently existing system. When a similar structure is used, the real system experts can reflect better to the functionality and correctness of the model and it is better recognisable. This fastens the search for deviations from the real system.

- Secondly, during the software design process, we were able to reflect – in terms of functional and technical performance – on the conceptual design decisions about software structures well before the implementation of the production software had started.

- Moreover, in an early stage we were able to link real software to the simulation model and test functional and technical performance of production software.

- Even more important, we were able to determine priorities for improvement measures, much faster than would have been possible when the improvements had had to be implemented in production software.

- Finally, the algorithms implemented in the simulation, provided a basis for the implementation going beyond functional specifications.

We even conclude that without an architecture of the model system similar to that of the production software, it would have been much more difficult to validate the model system for analysis purposes, and secondly, it would have been impossible to support software design beyond the testing of algorithms in functional terms. By applying a similar architecture, the robustness of the design has been improved without increasing the lead-time of the project.

We also experienced two problems regarding this guideline. First, because it is inevitable to have the model system earlier ready than the actual software, the architecture of the software needs to be ready when the model system is developed. In this case and most likely in other cases too, this is not feasible. Therefore, we started with a preliminary architecture and kept it up-to-date during the project, which required additional resources to do so. Secondly, the fact that the model system has to evolve continuously during the design and implementation process, leads to the impossibility to have it always ready for performance tests or other
functional testing, which sometimes delays the process. On the other hand, the model system supported the design of the architecture and the interface, which led to relatively small changes of the interface when the entire software architecture became ready.

7.7.4 Interaction automated system with manned operating tasks

The analysis of the current system – the system that is being replaced - shows that one of the problems is the interaction between the operators in the control tower and the automated system. First, the operators lack the possibility to intervene whenever an unexpected event occurs. The result of the inability to react quickly is that small errors lead to larger disturbances than necessary. Secondly, the feedback from the automated system to the manual configuration of the system takes time, whereas the operators only assess the short-term effects. In a number of cases, this has led to a worse performance on the longer term. Thirdly, the operator can make mistakes, which lead to a poorly performing automated system. Therefore, the application of the following guidelines could improve the design of the system:

− The operator should have a maximum possibility to intervene the automated operation in the event of unexpected situations. In order to determine which intervening possibilities an operator should have, an analysis should be made of events that can occur, which are not covered by the automated system. Furthermore, the basic functions like assigning a job to a piece of equipment, or assigning a specific move to a piece of equipment should be standard interventions that a manual operator can perform.

− The possibilities to configure the system by the operator should be limited to a minimum; generally, the automated system should be capable of configuring itself. Moreover, the input of the operator into the system should be checked and monitored by the automated system, so that the operator can get the feedback he needs to assess his configuration. The operator should be advised by the automated system about decisions that could affect the performance of the entire system.

7.7.5 Interaction between manual and automated operations is a key factor

In the design process of the new PCS, extensive attention is paid to the interaction between the people that the operators of the PCS and the system itself. We found
that in the old-PCS, the operators got too little feedback about their interventions, which made manual intervention cumbersome. Therefore, the following measures are taken:

- Many tasks that were formerly done by the operator will be automated, because of their complexity. However, the operator will have the possibility to correct the system in the case he thinks something should be changed. This is the case when the controller has information the system does not have available.

- More detailed output and feedback will be generated by the system on-line. This feedback should create a faster learning loop for the user to evaluate his actions.

Furthermore, operational experts and future system users are involved in the design project from an early point on to avoid misunderstanding of the solutions and to improve the controllability of the system.

7.7.6 Performance measurement should be part of the PCS

A system as complex as an automated container terminal requires a detailed set of instruments for measuring the behaviour of individual components as well as the result of the interaction of these components, i.e. the system behaviour. We experienced here that it is sometimes difficult to get circumstances that are easy measurable in practice, whereas in the model system it is easy to measure and to create circumstances in which measurements make sense. Difficulties are for instance:

- To create comparable conditions. Because measuring is meant to compare situations, the situations need to be comparable in the first place. In a model environment such comparable conditions can be created easily because the conditions can be set by the user. In practice, no operation is the same as any other, which makes comparison more difficult and therefore makes assessment based on the measurement difficult.

- To determine cause and effect. In a system with human operators and drivers, a similar measurement can be a result of a complex of factors; in many cases one should also measure underlying parameters to assess the cause. An example is the quay crane productivity: it may be that a certain productivity
level is caused by (1) a slow crane, (2) a slow crane driver, (3) long moves into the vessel, or (4) lack of container supply to the crane. Of course, all of these effects can be quantified, but the example makes clear that it is insufficient to measure quay crane productivity and assess the quality of the process control system by this performance indicator.

During the design, such a set of instruments should be an integral part of the solution in order to avoid later problems with the addition of these tools. In the design of the new PCS, this guideline is applied in the sense that:

- Extensive attention is paid to measuring performance during the design, in particular by means of simulation, but also by designing a feasible performance measurement method into the real system and into the equipment.

- Many measurements of the old situations were performed by means of a tool that interprets logged events into statistics, e.g. regarding failures, waiting times, cycle times. This tool will become an integral part of the new software.
8 Application: Design of an Automated Stacking Crane

In the previous chapter, a case study has been discussed that comprised an entire terminal redesign study. The case discussed in this chapter goes into far more detail, which shows that our proposed approach is not only applicable at the level of an entire terminal, but also when answering more detailed (technical) questions. It describes our experiences with the application of our approach in the design of an automated stacking crane.

8.1 The automated stacking crane as necessary new technology

In this chapter, we discuss a handling system that is one of the possibilities for terminals that are searching for a high density storage solution in combination with a reduction of labour costs. In detail we will analyse a piece of semi-automated equipment – an overhead bridge crane - that can be used at container terminals for yard operations. The motivation for introducing this solution is the scarcity of land in port areas, which increasingly forms a limitation for expanding ports, and in particular container terminals. Especially in Europe, the availability of land is a major issue for container terminals. Existing terminals are often located in urban areas where extension is only possible by means of reclaiming land. As this is a rather expensive and sometime long lasting approach, terminals are trying to increase their throughput per hectare by increasing the stack density.

Figure 8-1: Typical surroundings of a terminal in Europe coping with a stacking system with low density and confronted with a large increase in volume (Port of Piraeus, Greece).
Increasing the stack density can be done in various ways. A straightforward way is by increasing the stacking height. However, the disadvantage of an increasing stacking height is the increase of rehandles (also called shifters, or shuffle moves) that are needed to deliver the right container because other containers are standing on top of it (see Figure 8-2). Especially at the landside, the quality and timeliness of information is poor, which makes it impossible to prepare the yard avoiding shuffling. This will result in a high percentage of additional moves that has to be executed to deliver the right container.

![Percentage rehandling as percentage of total moves](image)

*Figure 8-2: Effect of an increasing stacking height on the percentage unproductive moves in the case of random stacking, which is usually the case at the landside truck operation.*

An alternative is to increase the stacking density by limiting the percentage area used for access roads, equipment, and so on. However, this limits the possibilities of applying certain handling systems. As can be seen in Figure 4-12, the density of different stacking systems differs significantly. A system combining a high density and medium height – to limit the effect of the lack of information - seems to be a feasible approach. Therefore, we will look in detail into a piece of yard handling equipment combining these characteristics.

Two systems that have a *low* stack density but are widely spread are the straddle carrier operation (especially in Europe), and the wheeled stack (especially in the United States). Although these handling systems provide a high flexibility and
relatively low amount of equipment, the density is much lower than of RTG, RMG or OBC solutions. The stacking system with the highest density is the overhead bridge crane (OBC) – see Figure 8-3. Due to the construction of concrete piles, the area lost for other things than stacking containers can be kept to a minimum.

<table>
<thead>
<tr>
<th>State-of-the-art RTG</th>
<th>RMG-design (1)</th>
<th>RMG-design (2)</th>
<th>OBC-design</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>119%</td>
<td>86%</td>
<td>138%</td>
</tr>
</tbody>
</table>

Figure 8-3: Four different yard handling systems and their relative density compared to an RTG

Such a crane is currently used at the Pasir Panjang terminal of PSA in Singapore (relative density: 129%) and is under consideration at a container terminal to be built in Antwerp (see Figure 8-4). The two designs, however, are quite different. The OBCs in Singapore are eight high, 14 wide with truck roads under the OBC. The truck travels to the OBC, which allows the OBC to be rather slow (2 m/s). The OBCs planned to be realised in Antwerp are four high, nine wide, with interchange at the top and end of the stack; the OBCs are travelling to the transfer points, and therefore they have to be fast (4 m/s). Both cranes are semi-automated: within the stack they operated unmanned, when interchanging with a manned truck, a remote operator performs the fine positioning.

Figure 8-4: Two different OBC designs: left the design under consideration for a terminal in Antwerp, right the design implemented in Singapore.

Chapter 8: Application: Design of an automated stacking crane
The OBCs that are planned to be realised in Antwerp fit in a system with shuttle carriers (1 over 1 straddle carriers) for the waterside transportation, and direct truck handling at the landside ends of the stack by the OBC (see Figure 8-5). The transportation to the rail facility is either by means of shuttle carriers or a tractor-trailer system, similar to the solution at CTA in Hamburg. Per stacking module – perpendicular to the quay as is shown in Figure 8-5, there will be two OBCs installed: one to operate the waterside, one to operate the landside. The average stacking density of the OBC system is approximately 1500 TEU / ha, which is two to three times as high as a straddle carrier operation with 1 over 2 straddle carriers.

Figure 8-5: Layout with OBCs and shuttle carriers (source: Gottwald Port Technology, 2003)

8.2 Scope, objective and approach of the project

The objective of the project subject in this chapter is to design an OBC that loads at both ends of the stack that fulfils specific productivity demands and can be built as cost-efficient as possible. The latter is of importance because the competition at the market for stacking cranes is fierce.

Because the OBC is semi-automated, not only the design of the hardware of the crane (e.g. structure, drives, controls, power supply) is of relevance, but the design of the control software is as important. Therefore, both the hardware and software design are considered.
In the design process, we apply a simulation approach, starting with the conceptual model of the system, i.e. the stacking crane. Then we create a model system, which will be used for experimentation with various configurations. We will base the conclusions regarding the design on the outcome of these experiments.

**Figure 8-6: Pilot-plant OBC in Antwerp (courtesy of Gottwald Port Technology, 2002)**

### 8.3 Conceptual model of the stacking crane

#### 8.3.1 Hardware

The OBC system concept consists of six main components (see Figure 8-6):

1. Two elevated crane tracks, made of concrete (or steel) with rails on top. The concrete (or steel) pillars are under piled. The crane tracks have a variable length depending on the demand for storage capacity.

2. Two or more interchange zones. Because the crane track is elevated, neither the rails nor the crane interferes with transportation equipment on the ground. Therefore, multiple interchange zones can be constructed. In this case, there are two interchange zones, one on the waterside, and one on the landside of the stacking module.
3. One or more cranes (gantries). The track is able to carry multiple cranes. In this case, two cranes are used. One for each side, although both cranes are able to serve the entire stack. The crane travels up and down the stack module.

4. Each crane is equipped with a trolley, moving perpendicular to the crane.

5. Each crane is equipped with a hoist, performing the hoisting and lowering movement.

6. Each crane is equipped with a spreader, performing the grab and drop action and adjusting its length to fit the actual container length.

Apart from the stack density, the OBC-concept has another beneficial property. Because the crane does not have legs – most other stack handling equipment has legs, like a straddle carrier, an RTG or an RMG – it can move faster without skewing problems (up to 4.5 m/s). Another advantageous property is the lack of interference with the transportation on the ground; most stack handling systems have – due to the crane legs or rails – interference with the transportation system, whereas the elevated crane does not have this problem.

A major disadvantage of an elevated rail track is the need for piling, which is more expensive than a track at ground level. Most stack handling systems do not need additional preparations except for rails on sleepers. The under piled elevated rail track (± € 5,000/m, source: Gottwald Port Technology, 2002) may be up to four times as expensive as an RMG on rails with sleepers (± € 1,350/m, source ECT, Gottwald Port Technology, 2002). Besides, it lacks the flexibility of an RTG or straddle carrier which can be moved from one stack area to another.

Important issues regarding the hardware design are the following:

- The number of drives to be installed; the question is whether two or three converter sets for drives are required; in the first case, one converter is switched between gantry and hoist.

- The power of the drives, resulting in the speed and acceleration of the gantry, trolley, and hoist.

- The energy supply that is required to deliver enough power to an automated yard equipped with OBCs.
The acceptable lateral forces on the crane track, mainly influenced by the simultaneous movements of crane and trolley with heavy loads and/or under windy conditions (up to 10 Beaufort).

8.3.2 Software

The software that controls the (semi-automated) OBC is as important as the hardware, especially because there are two cranes on one track. Therefore, the software not only controls the crane movements, but also the collision avoidance, the interchange protocol at both interchanges and the container locations within the stack. The cranes receive their jobs from the process control system, containing a job assignment component. The interaction between the crane control software and the process control system is important to make the cranes productive. In Figure 8-7, the structure of such crane control software is shown.

Figure 8-7: Layered control software controlling the execution of the OBC

It is the task of the crane control software to enable all functionalities provided by the hardware and make sure that the cranes will execute their jobs according to the plan. This means that for instance deadlock situations between the two cranes have to be solved by the crane control software.

It is the task of the high-level control software (TOS) to assign jobs to the OBC in such a way that the crane can work efficiently. Since the low level control is only responsible for the execution of the jobs, the performance of a stacking module mainly depends on the high level control.
8.4 Creating a model system of a stacking crane

8.4.1 Scope of the model system

In order to support design decisions regarding the hard- and software of the OBC a model system is developed containing all relevant aspects for the design project. The model comprises a single stack module, with two cranes on one track. The structure of the model system mirrors the control software architecture depicted in Figure 8-7, which means that an object-oriented architecture has been applied. Besides, the model system contains a component representing the process control system, generating jobs for both the cranes. The following aspects are considered within the model system:

- Load dependent hoisting and lowering.
- Wind speed dependent crane driving.
- Simultaneous movement of all equipment, i.e. crane, trolley, and hoist (only possible with three converters).
- Limitations to simultaneous driving.
- Collision control able to solve deadlocks.
- Hoist movement with slow positioning.
- Movement patterns with possibility to move with the trolley at the transport position as well as direct movements where the trolley is not in the centre of the rail track (see Figure 8-8).
- Claiming of interchange in order to avoid collisions with entering straddle carriers or other equipment.

Furthermore, all parameters in the model system concerning the crane kinematics are user configurable, so that it is possible to vary and test the effect on the productivity.
8.4.2 Verification and validation

In order to verify and validate the model system, we made a comparison between the model system and the pilot plant of the OBC. The model and the pilot crane are compared using the same moves, with the same container types and container weights. The varying engage and disengage times, usually modelled as a stochastic distribution (uniform distribution between 1 and 5 seconds) within the simulation model, are replaced by the actually measured times (approximately 2 seconds) to exclude this influence. The validation, made in multiple iteration loops, shows that the maximum difference within one complete crane cycle is one second. This deviation is very small compared to the average crane cycle time, which is approximately 180 seconds. Besides, we developed a visualisation of the movements by means of a true to scale 3D animation, in order to follow the movements of each machine, i.e. gantry, trolley, hoist. The firstly developed true to scale 2D model was considered to be insufficient to validate all movements, because the movements take place partly simultaneously. Although registration of movements in time-way diagrams is present, a detailed visualisation helps to verify the model. The visualisation is depicted in Figure 8-9.
8.5 Model treatment

8.5.1 Alternative configurations (scenarios)

The design alternatives that are present contain the following degrees of freedom:

- The length, width, and height of the stack module. Depending on the location, the dimensions will be different, based on a number of factors. In this specific case, the width and height are not varied, but the length is.

- The number of converter sets in the crane. Either each piece of equipment has a separate converter set, or the converter has to be shared between the pieces of equipment. The latter solution causes switching delays, but saves the cost of a converter set.

- The speed and acceleration of the gantry, trolley, and hoist. In order to increase the performance, the speed and acceleration can be increased by increasing the power of the drives. Of course, the more powerful a drive is, the more costly. Furthermore, the hoisting speed is load dependent, which means that the heavier the container, the slower the container can be hoisted or lowered.

- The degree to which the crane and trolley can move simultaneously. Because the crane travels above the containers, the movement in X and Y direction could be executed at the same time to reduce the duration of a move. However, simultaneous movements increase the forces on the crane track, especially with high loads or heavy wind loads. The further the simultaneous
movement outside the centre of the crane track, the higher the forces. Therefore, to reduce the cost of the crane track, one could limit the possibilities for simultaneous driving, eventually even load dependent.

- Optimisation of the collision control between the two cranes. Because the two cranes interact on the same crane track, the collision avoidance mechanism could influence the crane performance. Therefore, two collision avoidance algorithms were considered.

![Figure 8-10](image)

*Figure 8-10: Division of stack module in areas; from these areas random locations are drawn to create the moves as indicated.*

All configurations are tested under the same operational conditions, i.e. a continuous operation of 16 hours with constant conditions apart from the specific treatments that are applied. The conditions are summarised in the tables at page 212. Figure 8-11 shows the frequency distribution of move types that each crane performs. Each move is randomly selected from this list, according to the relative frequencies as given. The meaning of each move is shown in Figure 8-10. Subsequently, the configuration of the load dependent hoist movement is given (see Figure 8-12). The distribution of container weights is also shown (see Figure 8-14). Furthermore, the dead times in the crane movement (see also Figure 8-8) are depicted (Figure 8-13). Finally some other relevant configuration parameters are shown (see Figure 8-14).
Figure 8-11: Frequency of type of moves executed by each crane

<table>
<thead>
<tr>
<th>Waterside crane</th>
<th>Type</th>
<th>Scenario 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterside Interchange -&gt; Waterside stack</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>Waterside Interchange -&gt; Intermediary stack</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>Waterside stack -&gt; Waterside Interchange</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>Intermediary stack -&gt; Waterside Interchange</td>
<td>4</td>
<td>10%</td>
</tr>
<tr>
<td>Shuffle in waterside or intermediary stack</td>
<td>5</td>
<td>20%</td>
</tr>
<tr>
<td>Waterside Interchange -&gt; Landside stack</td>
<td>6</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Landside crane</th>
<th>Type</th>
<th>Scenario 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landside interchange -&gt; Landside stack</td>
<td>1</td>
<td>15%</td>
</tr>
<tr>
<td>Landside interchange -&gt; waterside stack</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>Landside interchange -&gt; Intermediary stack</td>
<td>3</td>
<td>15%</td>
</tr>
<tr>
<td>Landside stack -&gt; Landside interchange</td>
<td>4</td>
<td>15%</td>
</tr>
<tr>
<td>Intermediary stack -&gt; Landside interchange</td>
<td>5</td>
<td>15%</td>
</tr>
<tr>
<td>Landside stack &lt;-&gt; Waterside stack</td>
<td>6</td>
<td>30%</td>
</tr>
<tr>
<td>Shuffle in landside or intermediary stack</td>
<td>7</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8-12: Load dependency of hoist speed

<table>
<thead>
<tr>
<th>Container Load (ton)</th>
<th>Speed (m/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>2</td>
<td>60.0</td>
</tr>
<tr>
<td>5</td>
<td>60.0</td>
</tr>
<tr>
<td>10</td>
<td>60.0</td>
</tr>
<tr>
<td>15</td>
<td>59.5</td>
</tr>
<tr>
<td>20</td>
<td>54.0</td>
</tr>
<tr>
<td>25</td>
<td>48.5</td>
</tr>
<tr>
<td>30</td>
<td>43.5</td>
</tr>
<tr>
<td>35</td>
<td>38.0</td>
</tr>
<tr>
<td>40</td>
<td>30.0</td>
</tr>
</tbody>
</table>

Figure 8-13: Dead times of machines

<table>
<thead>
<tr>
<th>(s)</th>
<th>gantry</th>
<th>trolley</th>
<th>spreader</th>
<th>drop</th>
<th>grab</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>1.0</td>
<td>1.0</td>
<td>3.0</td>
<td>0.0</td>
<td>1.6</td>
</tr>
<tr>
<td>end</td>
<td>1.0</td>
<td>1.0</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 8-14: Other parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interchange time waterside (s)</td>
<td>0</td>
</tr>
<tr>
<td>Interchange time landside (s)</td>
<td>30</td>
</tr>
<tr>
<td>Length of track (TEU)</td>
<td>34</td>
</tr>
<tr>
<td>Distance to interchange (m)</td>
<td>15</td>
</tr>
<tr>
<td>Width per TEU slot (m)</td>
<td>2.85</td>
</tr>
<tr>
<td>Length per TEU slot (m)</td>
<td>6.50</td>
</tr>
<tr>
<td>Height per TEU slot (m)</td>
<td>2.59</td>
</tr>
<tr>
<td>Jog height loaded (m)</td>
<td>0.50</td>
</tr>
<tr>
<td>Jog height empty (m)</td>
<td>0.40</td>
</tr>
<tr>
<td>Jog speed (m/s)</td>
<td>0.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weight Class</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>25%</td>
</tr>
<tr>
<td>10</td>
<td>20%</td>
</tr>
<tr>
<td>20</td>
<td>20%</td>
</tr>
<tr>
<td>30</td>
<td>20%</td>
</tr>
<tr>
<td>40</td>
<td>15%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>16.00</strong></td>
</tr>
</tbody>
</table>

The jog speed and distance are used for the weighing of the container and the check for the correct engagement (during hoisting) of the twistlocks, and for fine positioning (during lowering).
There is no start-up time and all results are based on a single replication. For all treatments common random numbers have been applied reducing the number of replications required to achieve comparable results. To verify whether one replication is sufficient, we carried out one experiment with 8 replications without common random numbers to test the variation in outcomes. As can be seen in Figure 8-15, the results of different experiments vary slightly – 95% interval of 1.2 moves around an average of 19.6. Because consecutive moves do not influence each other, the run length of 16 hours is assumed to be sufficient to get valid results.

<table>
<thead>
<tr>
<th>Hour</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
<td>16</td>
<td>13</td>
</tr>
</tbody>
</table>

Figure 8-15: Results of 8 replications with one configuration. Each cell represents an individual measurement per crane (C1 is waterside crane, C2 is the landside crane). The setting of the crane: gantry speed = 4 m/s, trolley speed = 1 m/s, hoist speed (max) = 1 m/s.

8.5.2 Results

In the following diagrams, the comparison of the various alternative configurations is depicted. In Figure 8-16, the effect of the length of the stack module is shown. Of course, the effect is heavily dependent on the origin and destination of containers, but the list of moves represents a realistic pattern. As we can see in the diagram, the effect in this scenario is relatively limited: the productivity per crane drops with two moves per hour when the length is increased from 34 TEU (221 m) to 54 TEU (351 m). This can be explained by the fact that there are two cranes that both serve one end, and by the high speed of the crane: travelling the entire length takes less than 1.5 minutes in the case of a 54 TEU long stack.
Chapter 8: Application: Design of an automated stacking crane

Effect of the stack length

2-8 Corridor - Vgantry = 4.0 m/s - Agantry = 0.3 m/s2 - Vtrolley = 1.0 m/s - Atrolley = 0.3 m/s2 - Vhoist = 1.0 m/s - Ahoist = 0.33 m/s2 - Scenario 3

Figure 8-16: Effect of the length of the stack module

In Figure 8-17, the effect of an additional (third) drive is shown. When there are just two converter sets, the crane needs to switch between converters when starting the lowering or hoisting. This switching costs time, up to 3 seconds per switch. For an entire cycle (pick and put container) this time can add up to 15 seconds. The cost of an additional converter set is approximately 30 thousand Euros per crane. In relation to the performance benefit (up to 1.2 mph), this might be worthwhile to consider.

Effect of separate converter set for hoist

Figure 8-17: Effect of an additional converter set for simultaneous gantry driving and hoisting
Effect of gantry speed

$A_{\text{gantry}} = 0.3 \text{ m/s}^2 \quad V_{\text{trolley}} = 1.0 \text{ m/s} \quad A_{\text{trolley}} = 0.3 \text{ m/s}^2 \quad V_{\text{hoist}} = 1.0 \text{ m/s} \quad A_{\text{hoist}} = 0.33 \text{ m/s}^2$

Figure 8-18: Effect of the gantry travelling speed on the productivity

In Figure 8-18 and Figure 8-19, the effect of varying the gantry characteristics is shown. Interestingly is that although the crane track is fairly long (251 m from interchange to interchange in the case of 34 TEU), the effect of the gantry speed is insignificant. When increasing the maximum speed from 3.0 m/s to 5.0 m/s the productivity increases with a maximum of 7% for the landside crane, whereas the benefit for the waterside crane is 3%. The effect of an increase of the acceleration is slightly higher, but still very limited. Especially to achieve higher acceleration numbers, the traction of steel wheels on steel rails has to be improved, which requires additional provisions. When increasing the acceleration with 50% (from 0.3 m/s$^2$ to 0.45 m/s$^2$) a productivity benefit of 5% is achieved. In comparison to the additional strength of the crane track that is needed to support a crane moving at speeds higher than 4 m/s, the productivity increase is small, especially when the costs are considered.
The crane is designed allowing for simultaneous gantry and trolley movement. Since the gantry movement takes in most cases longer than the trolley movement, the trolley movement is hardly ever critical. However, simultaneous movements put additional forces on the elevated crane track and therefore the limitation of simultaneous driving may significantly save costs of the crane track. As can be seen in Figure 8-20, the possibility to drive simultaneously with gantry and trolley is critical for the crane productivity, even in the case of a short stack where already in a proportionally large share of moves the trolley movement is critical. The difference between no limitation, i.e. at all nine rows the gantry and trolley are allowed to drive simultaneously, and a limitation that only allows for driving in the middle of the crane track, is 3.5 to 5.0 mph, or 20%-25%. The limited driving at row two and eight already decreases performance with 7%. Therefore, we conclude that the simultaneous driving is one of the critical factors for the productivity of an (automated) stacking crane with front-end service. This conclusion is confirmed by an earlier study (Dobner et al., 2002a). Here we also found a productivity benefit of 20%-25%.

Figure 8-19: Effect of the gantry acceleration/deceleration
The next treatment concerns the software algorithm for the anti-collision between the two cranes. Because both cranes serve the entire stack, and operate where the workload is, collision avoidance is required. Not only collisions should be avoided, also deadlocks should be solved. The two collision avoidance algorithms that are compared are based on the claiming of a certain area in the stack. The difference is the time at which the area is claimed. In the first algorithm, the area is claimed as soon as the gantry movement is started. The gantry tries to claim the entire length to his destination position. If it does not succeed, it decides whether to wait or to move to the position nearest to the destination that can be claimed. The gantry will wait in the case the other crane moves in its direction. In the case the other crane moves away, the gantry will drive to the nearest position and wait until the other crane reduces its claim.

The second algorithm is a first-win algorithm. The gantries claim a minimal distance in advance (based on the braking distance) and the crane that arrives first is the winner; the other crane has to wait. Although this algorithm sometimes leads to moving up and down the stack, because the crane is sent away by the “winner” crane, it theoretically leads to the fastest moves. In practice, there is no significant difference between the two algorithms in terms of performance (see Figure 8-21).

Figure 8-20: Effect of limiting the simultaneous driving of gantry and trolley
Figure 8-21: Effect of claiming algorithm

Besides the alternatives that are considered for optimising the crane, we analysed the impact of the job assignment to the crane and the effect of the yard strategy on the crane productivity. The result is shown in Figure 8-22. Two movement patterns are compared for the scenario that was likely to occur in a life situation (scenario 1 is the standard configuration, used for all the other experiments) with the so-called acceptance test. The types of moves executed by both of the cranes are enlisted in Table 8-1.

Table 8-1: Overview of moves of cranes per scenario

<table>
<thead>
<tr>
<th>Waterside crane</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterside Interchange -&gt; Waterside stack</td>
<td>20%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Waterside Interchange -&gt; Intermediary stack</td>
<td>20%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Waterside stack -&gt; Waterside Interchange</td>
<td>30%</td>
<td>25%</td>
<td>33%</td>
</tr>
<tr>
<td>Intermediary stack -&gt; Waterside Interchange</td>
<td>10%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>Shuffle in waterside or intermediary stack</td>
<td>20%</td>
<td>40%</td>
<td>33%</td>
</tr>
<tr>
<td>Waterside Interchange -&gt; Landside stack</td>
<td>0%</td>
<td>0%</td>
<td>33%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Landside crane</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landside interchange -&gt; Landside stack</td>
<td>15%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Landside interchange -&gt; waterside stack</td>
<td>0%</td>
<td>0%</td>
<td>33%</td>
</tr>
<tr>
<td>Landside interchange -&gt; Intermediary stack</td>
<td>15%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Landside stack -&gt; Landside interchange</td>
<td>15%</td>
<td>15%</td>
<td>33%</td>
</tr>
<tr>
<td>Intermediary stack -&gt; Landside interchange</td>
<td>15%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Landside stack &lt;-&gt; Waterside stack</td>
<td>30%</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>Shuffle in landside or intermediary stack</td>
<td>10%</td>
<td>10%</td>
<td>33%</td>
</tr>
</tbody>
</table>
The difference between the two real-life scenarios is the percentage unproductive moves – moves that do not begin or end at an interchange - that have to be executed; 20% in the first scenario, 40% in the second for the waterside crane; the landside crane does 40% unproductive moves in both scenarios. In the acceptance test scenario, the percentage of unproductive moves is 33% for both cranes. Moreover, the productive moves take place at the other side of the stack, hindering the other crane to a maximum. Based on the results, we can conclude that the percentage of unproductive moves has a significant impact on the performance.

Figure 8-22: Effect of shuffling and housekeeping on crane productivity

In all scenarios, the landside crane is busy with a large percentage of unproductive moves. This is a scenario that is common in the case of two cranes on a single track. The waterside crane has to cope with high peaks: the landside crane is supporting the waterside crane in most of the times. Only during rush hour on the landside, the landside crane might be fully utilised by landside jobs. As can be seen in Figure 8-22, the waterside crane shows a productivity of 15.4 moves per hour (mph) in the acceptance test scenario, 19.9 mph (+29%) in the scenario with 40% shuffling and 23.0 mph (+49%) in the case of “only” 20% shuffling.

The experiment shows the impact of unproductive moves on the productivity of a machine. The cause of unproductive moves may be missing information, changing
information or bad planning. Therefore, this experiment shows the interaction between high level planning and execution of orders at crane level.

8.5.3 Conclusions regarding crane optimisation

Based on the experiments, we conclude that there are two things that have priority when optimising yard crane productivity. First, the co-operation between the yard planning, job assignment and crane design, which is reflected in the move schedule of the cranes. In the case, for instance that the information provision is of low quality, many shuffles will be required, which requires a fast hoisting movement rather than a fast gantry or trolley movement. Secondly, the simultaneous driving of gantry and trolley is a way to increase productivity significantly. All other configurations have insignificant or minor effects on the crane performance. Furthermore, analysis with this type of cranes as part of an entire terminal has shown (see e.g. Dobner et al., 2004) that in most of the time, there is time left between the moves. Even in the peaks, idle times of on average 5-15% occur. Although these idle times are scattered over the hours, the small improvements that faster cranes bring, would not increase the productivity of the entire system. Therefore, it remains important to verify the conclusions of isolated experiments – as presented in this chapter – in the context of an entire terminal operation.

8.6 Findings and conclusions

In this section, the experiences with the application of our proposed guidelines in the design project of a semi-automated yard crane are discussed. In chapter 9, we reflect from a broader point of view on the observations from this and the previous chapter.

8.6.1 A concurrent simulation approach leads to more cost-efficient design

Many different configurations have been tested by means of simulation, whereas the testing with the real crane would have taken a lot of time. Besides, the crane has also been used in more comprehensive model systems to provide even more feedback to the design issues. This would not have been possible with real cranes. Furthermore, because the model system depicted the processes in the crane at a detailed level, the model could be used for detailed design questions. By running various scenarios, priorities for design could be set and the crane could be designed in a more cost-effective way. An example is the high value that initially was attributed to the high
gantry speed of 4 m/s, which proved to be of less significance with the planned length of the stack module.

Although not yet realised, the model could serve as on-line test environment for the yard planning and job assignment software. Because the model runs fast and multiple modules can be used at the same time, a stack can be simulated. This also allows for long term experimentation with the model, such as evaluating the yard planning (grounding strategy) on the middle-long term. This can avoid later problems like high shuffling percentages or unnecessary housekeeping moves.

8.6.2 Holistic approach is required to determine realistic test scenarios
In order to test a single component of a bigger system under situations that may occur in a terminal operation, the model of the automated stacking crane was plugged into an entire terminal model. There, the moves executed were logged and used as real-life test scenarios for the crane. Although in an operation with multiple stack modules, there may be idle time between the moves – resulting in average productivities of a stacking crane of 8-12 productive moves per hour – the origin and destination of the moves are similar in both operations, and therefore considered valid for designing the yard crane.

The preferred test environment of the crane is therefore the entire terminal. Only then, the eventual benefit of a faster crane can be observed. Experience shows that then even more, the small benefits that showed up during the isolated experiments, are not leading to a significantly higher terminal productivity. More is gained by controlling the cranes in an intelligent way, which is reflected in the move schedule and the interaction with the waterside and landside transportation system. For tuning and validating the model system, however, an isolated setting is most convenient.

This supports our guideline to use a holistic approach rather than an approach where individual components are designed in isolation. Although the treatments with various types of move lists provided insight in the effect of the high level planning, it is an approach that is insufficient to design a crane in such a way that the performance of the entire terminal operation can be maximised.

8.6.3 Object-oriented simulation reduces algorithm development time
As mentioned above, a model system of the stacking crane has been developed as component. After the model had become ready, it was plugged into the total terminal
model system representing the entire terminal operation (see section 4.2.8) to operate there as one of the components. The fact that the model system has been built in an object-oriented way, made it easy to create N instances of the components and integrate them into the terminal model. Nevertheless, not only for this purpose, object-orientation proved to be useful. The internal structure of the model (see Figure 8-7), allows for an easy extension of the model to a model with more cranes by adding crane objects to the model. Because it is likely that all cranes within one stack module are equal, they simply inherit all configurations from their mother object. In the case the crane configuration changed, only the mother object had to be altered.

Finally, the object-oriented structure made the algorithms for the collision avoidance very similar to the real production code (also object-oriented), which made comparison and validation easier.

8.6.4 Integrated hard- and software design to accomplish cost-efficient solutions

The results of the model treatments show that the planning and yard grounding strategy – represented in the type of moves that each crane had to perform – have a large impact on the productivity of the stacking crane. When cranes have to execute many unproductive moves the potential for performing productive moves decreases. Furthermore, the job assignment of cranes should consider the disturbance between the cranes. In the acceptance test scenario, the total percentage of unproductive moves was lower than in the other scenarios, but the productivity was lower, because the cranes disturbed each other very frequently.

Moreover, these aspects cannot be solved by making the gantry speed faster, for instance. They require an approach that addresses the hardware specifications and software specification as integrated design issue, also considering the actual conditions at the site where the crane will be applied.

8.6.5 Concluding remarks

This chapter reflects the design of a semi-automated yard crane, as we have experienced it in many terminal design projects. The future terminal operators sends out a specification for a yard crane, and the potential suppliers design the crane in isolation, not knowing what kind of conditions have to be met in practice. In our opinion, this approach cannot lead to an efficient nor effective crane design seen
from a holistic perspective, taking the terminal’s performance targets as overall aim. It is our experience that taking the component as part of the entire system, the contribution of specific characteristics – for instance the ability to move in three axis at the same time – can only be assessed when the various designs are compared as part of an entire operation. When in these circumstances, the system is stressed, i.e. the highest workload expected is demanded, the effect of specific design features can be determined. This approach we propose to achieve a better balance between the technical qualities of a piece of equipment and the requirements it has to fulfil during a regular (peak) operation.
9 Reflection, Conclusions and Outlook

Chapter 9, the piece the resistance reflects on the findings when looking back to the research objectives and questions, and looking ahead to the future. It completes the picture by presenting the feedback we received from 10 experts in the field of container terminal design and operation.

9.1 Experiences with a simulation approach throughout the design process

In our research we explored the design of robotized container terminals. We started with the observation that the operation and design of robotized terminals differ quite drastically from traditionally operated terminals. The increased importance of software, but also the reliance on software makes the design of a robotized (automated) terminal different from the design of a manned terminal. On the one hand automation creates opportunities to improve the efficiency of the terminal – reduction of labour costs -, on the other hand it may limit the flexibility of the terminal. In practice, automated terminals are not yet seen as fully capable of achieving the same performance as manned terminals.

These observations led to the formulation of the following research objective:

To develop an approach for designing robotized, marine container terminals, which addresses the specific characteristics of such a terminal, and considers the specific properties of a terminal environment.

We pursued this research objective by carrying out a literature research into the topic of the design of container terminals, by exploring the problem of designing automated terminals by means of the analysis of the design and development process of the first automated terminal world-wide (ECT Delta terminal) (see chapter 2), by designing a framework of guidelines for designing automated terminals (see chapter 3), by developing the design approach that we aimed for, by designing a suite of models to support the design and decision-making process (see chapters 4, 5, and 6), and by applying and testing the design approach and the models in two action research studies (see chapters 7 and 8).

In this final chapter, we reflect upon the rigor, applicability, and relevance of our design approach by summarising the main findings (section 9.3, and by putting the approach into the perspective of the daily practice of a container terminal operator and/or people involved in container terminal (re)design. The latter we do by
presenting the results of an expert validation we carried out among ten specialists in this field (see section 9.2). Subsequently, we reflect from a more general perspective on the design approach proposed (section 9.4). We conclude this chapter with an outlook to future research (see section 9.5) related to this research.

9.2 Expert survey
The expert validation that we carried out consists of a remote survey that was performed by means of groupware software (Ventana, 2000). The survey consists of 17 questions that concern the guidelines we presented throughout chapters 3 to 6. The questions are listed in appendix 10.3. The survey is divided into two parts: the first part is meant to determine the respondent’s background with regard to simulation and container terminal design, as well as to the respondent’s experiences with the use of simulation; the second part contains the questions that reflect on the approach for designing robotized container terminals as proposed in this thesis.

9.2.1 Background respondents
There are 9 respondents out of 11 surveys sent out. The first question concerned the background of the respondent with regard to simulation. 3 respondents indicated that they are merely a user of simulation results to support decision making, 2 respondents answered that they used simulation results to support the analysis of systems, 4 indicated that they considered themselves a developer of simulation models, and none considered himself a developer of core simulation model components.

The second question asked the respondents to indicate for what purpose they use simulation. In the table below, an overview is given:

*Table 9-1: Use of simulation by respondents*

<table>
<thead>
<tr>
<th>Type of use of simulation</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support decision making</td>
<td>8</td>
</tr>
<tr>
<td>Create more insight into the systems I design or analyse</td>
<td>6</td>
</tr>
<tr>
<td>Visualise reality</td>
<td>3</td>
</tr>
<tr>
<td>Develop hardware or software</td>
<td>2</td>
</tr>
</tbody>
</table>
9.2.2 Answers to questions regarding simulation approach

Question 3 addressed the purpose of simulation and suggested a number of possible purposes, which could be assessed by the respondents. The respondents were asked to divide a maximum of 90 points over the 9 available purposes of simulation by importance of the purpose. The division of points looks as follows:

<table>
<thead>
<tr>
<th>Purpose of the use of simulation</th>
<th>Percentage of the divided points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare different alternatives</td>
<td>17%</td>
</tr>
<tr>
<td>Analyse systems that cannot be analysed using analytical tools or formulas</td>
<td>16%</td>
</tr>
<tr>
<td>Design an entire system, in this case a container terminal (e.g. terminal shape and size, handling system, logistic control system)</td>
<td>12%</td>
</tr>
<tr>
<td>Prototype control algorithms</td>
<td>12%</td>
</tr>
<tr>
<td>To convince my manager/client of the quality of a certain solution</td>
<td>10%</td>
</tr>
<tr>
<td>To visualise and animate the operational process</td>
<td>11%</td>
</tr>
<tr>
<td>Design equipment</td>
<td>9%</td>
</tr>
<tr>
<td>Test software</td>
<td>8%</td>
</tr>
<tr>
<td>Support day-to-day planning in the operation</td>
<td>5%</td>
</tr>
</tbody>
</table>

*Figure 9-1: Importance of various purposes of simulation*

Question 4 to 16 are statements that could be scored on a scale from 1 to 5, with 1 = strongly agree, 2 = agree, 3 = neutral, 4 = disagree, and 5 = strongly disagree. The average answer and the standard deviation per question are shown in the table below. The lower the average, the more the respondents agree with the statement. The lower the standard deviation, the more the respondents agree among each other.

From Figure 9-2, we learn that the respondents all agree on the upper six questions (highest average is 2.1). Also the standard deviation is low, which means that the variation around the average is low. There is one respondent disagreeing with the statement that switching between abstraction levels of model components contributes to the applicability of a model. Four respondents are neutral to this
statement. Interesting is that two of the respondents that reckon themselves to the simulation developers belong to the respondents answering neutral or disagree.

<table>
<thead>
<tr>
<th>Statement</th>
<th>avg</th>
<th>stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (robotised) container terminal design-engineering project would benefit from the use of simulation as support tool throughout the entire process, so from support of the early idea to final commissioning and operations.</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>In the case of an automated terminal, the design of software can benefit from the use of simulation when the architecture of the simulation model equals the architecture of the production software</td>
<td>1.5</td>
<td>0.8</td>
</tr>
<tr>
<td>In the case of the design-engineering process of a robotised container terminal, it is essential to represent the failure behaviour of equipment in the simulation environment, since it is an intrinsic property of the real system</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>The possibility to test production software components on-line by linking them to a simulation environment representing the entire environment of these software components reduces the time to develop software and improves the quality of the software</td>
<td>1.9</td>
<td>0.6</td>
</tr>
<tr>
<td>The use of a scaled 2D animation (what you see is what you get) in the design-engineering process of a container terminal, enables the user to understand the system better than a non-scaled animation does.</td>
<td>2.1</td>
<td>1.0</td>
</tr>
<tr>
<td>The possibility to switch between abstraction levels (levels of detail) within one model architecture, contributes to the creation of insight and to the applicability of simulation</td>
<td>2.4</td>
<td>1.2</td>
</tr>
<tr>
<td>The use (and re-use) of validated simulation components reduces the effort of validation of the entire model and contributes to the accreditation of a model:</td>
<td>2.4</td>
<td>0.8</td>
</tr>
<tr>
<td>The on-line linkage of production software components (for instance a job assignment component or an AGV router) to a simulation model, contributes to the validation and accreditation of a simulation</td>
<td>2.5</td>
<td>1.2</td>
</tr>
<tr>
<td>It is a new paradigm in systems development to use simulation as support tool throughout the entire design-engineering process</td>
<td>2.9</td>
<td>1.1</td>
</tr>
<tr>
<td>The use of a scaled 3D animation (what you see is what you get) in the design-engineering process of a container terminal, enables the user to understand the system better than a scaled 2D animation does.</td>
<td>3.1</td>
<td>0.6</td>
</tr>
<tr>
<td>A simulation is only useful when an abstract representation of reality (and not too much detail) is applied, because the essence of simulation is to provide a reduced representation of reality.</td>
<td>3.5</td>
<td>1.1</td>
</tr>
<tr>
<td>Simulation always takes too much time</td>
<td>3.8</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 9-2: Results of question 4 to 16

Interesting is also to see that 3D animation is not seen as strictly necessary to understand systems better than just with 2D animation. It contradicts with the huge efforts spent by the developers of simulation packages to provide 3D tools.

Finally, the respondents do not consider that simulation takes too much time. This slightly related to the statement concerning the level of detail. The respondents think that not only abstract representations are useful, but also detailed models. The latter category usually takes more development time before the experimentation can commence.
9.2.3 Findings and conclusions expert survey

Overall, the impression remains that the respondents predominantly agree with the philosophy behind the framework of guidelines proposed in this thesis. They clearly agree with the added value of using a simulation approach to support the decisions throughout the development process of a container terminal, and they confirm the benefits of using it to prototype and test software.

Still, the use of a simulation approach in a traditional way – supporting the assessment of alternative solutions - is seen as most important application. But topics as the prototyping of software, and the support of the entire design of a container terminal, including the logistical control system, are also seen as relevant applications.

Still, the support of day-to-day operations is not considered to be completely unimportant since still 8% of the points are allocated to this application, where the most important point – according to the respondents – got 17% of the points.

9.3 Research findings

In this section the main research findings are discussed. By answering the research questions as defined in chapter 2, we summarise the conclusions that can be drawn based on the applications of the design approach and the expert survey.

9.3.1 Findings with regard to the research questions

How can we enclose the (new) specific properties of processes at a maritime automated container terminal into the support environment that we use to gain insight from and perform the analyses with?

As we have selected simulation as main approach throughout the design-engineering of an automated container terminal, it is essential to capture the specific characteristics of this type of terminal into the models that are applied throughout the design process. As we have discussed in detail, one of the essential properties is the representation of the terminal operating system (TOS) within the models. When this is not accurately modelled, the model is hardly usable for the design of an automated terminal, because the quality of the TOS is a critical success factor. As we have learned from the process control software (PCS) replacement project at ECT (see chapter 7), the PCS (or TOS) determines the achievable service level of the
terminal to all modes of transportation. Representing the TOS of an automated terminal means – as compared to the representation of the TOS at manned terminals – the modelling of the software that controls the automated equipment, i.e. routing, collision avoidance, deadlock avoidance, et cetera. Because the degree of automation on behalf of the process control is higher at automated terminals, it requires the depiction of software components that perform the grounding strategy, plan the usage of equipment, plan and schedule the container movements, and assign the equipment to specific container movements.

Besides the representation of the functionality of the TOS, it is also important to represent the technical behaviour of the TOS, i.e. response times, asynchronous behaviour, and limited possibilities to optimize decisions.

*How can we reduce the risks of robotization and automation by means of a design approach?*

One of the risks of automation is to achieve a lower service level than planned. In the worst case, this leads to the inability to achieve the projected throughput volumes. It may also affect the attractiveness of the terminal, which may affect the profit. We have proposed an approach in which the achievement of performance is a major pillar. Throughout the design process, the application of a simulation approach enables the design team to verify whether the planned performance levels will be met considering the sensitivity for various assumptions with regard to process variables, such as cycle times, speeds, break-down occurrences, interchange times, et cetera.

A second risk of automation is that errors, especially in the TOS, lead to a terminal that does not run, i.e. complete stoppages, which is even worse than a performance loss. The approach we have proposed aims at reducing this risk by providing an environment that enables early testing of the software under more realistic conditions than traditional software testing methods provide. This is because of two elements: first the stochastic effects that lead to a far richer picture than any deterministic test scenario can depict; second the interaction between all components, which occurs because all components are tested in an integrated setting completely with all (modelled) non-software elements.

A third risk concerns the interaction between the TOS and the operator. Although we have not focused on this too much, the simulation approach allows for testing the
system including the interaction with an operator. It may even be strictly necessary for some applications, because these tasks in the TOS – for instance berth planning – are done by an operator instead of the software. An additional benefit of the combination TOS software, simulated environment, and real operator is the ability to train the operator with a realistic (i.e. dynamic) environment, rather than a static one, which does not contain the complex interactions, which may require operator intervention.

How can we ensure that the insight we provide is reliable and valid? How can we validate results when we do not have similar examples that are already operational?

Always when models are used to represent reality, the question should be asked whether the model is a valid representation of reality, given the scope of the analysis. In cases where the model depicts an existing system, validation is straightforward. However, in cases where a system of novel nature has to be built, especially on behalf of the TOS (for instance new optimisation algorithms), validation is a non-trivial process, if not impossible. As we have argued before, the TOS plays an important role for a container terminal, in particular when it is automated. Furthermore, we have observed that there are no standard TOS solutions for automated terminals. This means that the model of the TOS that will control the terminal to be built depicts something that does not exist yet. Here, the simulation model runs ahead of the music, meaning that the simulation is a prototype of the TOS. When the prototype deviates too much from the final TOS, the simulation may not be valid at all. To solve this, there are two alternatives. The first alternative is to work closely together with the software supplier that will deliver the TOS for the terminal. In this way, the simulated TOS can be built according to the architecture and functionality of the future TOS. The second alternative is to build the specification of the TOS according to the functionality of the simulated prototype. If then there is a reason for change during the development of the real TOS, the consequences should be assessed, and eventually the effect should be quantified by testing the change again by means of simulation.

We have followed both approaches in various cases (see chapter 7; also see Dobner et al, 2003b; 2004a). As we have reported in chapter 7, the complexity of the interaction between the real software design and the prototyping in a simulation...
model leads to deviations on both sides (i.e. prototype and software evolve concurrently), which leads again to tensions in time, i.e. performance assessments are expected although the design in neither the simulation model nor the real software is finalised. Handing over the specifications of the simulated TOS to the supplier of the TOS is the preferred approach if combined with the first approach, because the TOS supplier is likely to have already base software in place. This means that not all functionality can be implemented easily because the prototyped functionality may differ from the already implemented functionality in the software of the supplier. Therefore, the simulated TOS has to be updated according to the functionality of the software the supplier provides. As long as these changes are again verified on their impact to the performance of the terminal, the approach is still valid.

How can we ensure that the design approach is applicable to the portrayal and analysis of the new (innovated) processes, enabling new and old processes to be compared in the same formats and perhaps even driven by the same set of variables?

As we have argued above, modelling new systems – in particular on the software part – may be more complicated than modelling existing systems. Generally speaking, the design approach is unaffected by whatever system has to be designed and developed. The set of guidelines we have proposed (see section 3.4) appears valid to any container terminal design, although the exact contents will of course be determined by the individual setting of a specific design project. The latter will affect the set of variables used in the design process, but usually 80% of the variables will be similar to any container terminal design project.

On behalf of the model suite we developed, it may require additional model development in case of a completely new container terminal design. Currently we covered most of the common handling systems at container terminals, but as technology progresses, new systems may arise. However, the basic structure of the model environment is not likely to change, as the function of a container terminal is not likely to change either. Its storage and handling function, as well as its linking function between the various modes of transportation is not due to change, and therefore we think that the model environment will be able to depict future container terminal systems as well.
9.4 Reflection on the design approach

9.4.1 Experiences with the application of a simulation approach as ‘leitmotiv’

The central point of departure of our design approach – being the simulation (or problem-solving) approach applied throughout the entire design-engineering process – and also the point where our proposed approach deviates from what is common in (robotized) container terminal design trajectories, appears to contribute positively to the lead time of the design process and the quality of the end-product (i.e. the container terminal). It clearly contributes to the understanding of the solutions, it puts the contribution of specific solutions for terminal components into the perspective of the entire system – avoiding sub optimisation - and supports in separating the bad, average, and good solutions in an early stage of the process. Besides, it adds a new dimension to software design and testing as it provides a much richer (virtual) test environment of software components in an early stage of software design and development. Where traditional approaches use a simulation approach merely during the functional design – answering what-if questions – we propose to use simulation as approach for technical design and development support as well. The continued use of the models applied during functional design into subsequent steps in the design process, also continues the ideas at a greater level of detail (at the level of a working prototype) developed during earlier steps into the detailed design and implementation.

It is our experience that this approach is seldom used as far as the design of container terminals is concerned. Several reasons found during our research can explain why an approach that resolves a large part of the uncertainty that accompanies the design of a highly complex system, is rarely used:

- First, we experienced that simulation is still associated to a large extent with indicative sayings rather than accurate assessments, which would make them principally unsuited for precise tasks such as prototyping and/or testing of software. However, we argue this not to be true for the model systems that we developed during our research, which contain a high level of detail to represent operations at a container terminal with sufficient realism to provide the possibility to prototype software components. This is an approach that is – based literature review – not often followed (see e.g. Wysk, 2001), which also contributes to the fact that a simulation approach is not commonly used for purposes and/or decisions to be taken further in the design process.
Secondly, a simulation approach requires *additional time*, especially in the beginning of the project. This additional time is invested with the promise of a return on investment by means of better solutions that reduce the risk of the investment. However, the time needed, is not always available. Especially during the development of software, the time it takes to provide the feedback, may be longer than feasible, which leads to decisions made on perceptions rather than scientific analysis.

Thirdly, the professional environment within container terminals can be characterised as primarily focussed on *operations*. Although we observe a trend of an increasing level of education within the managerial staff of container terminals, many positions are still occupied by people that come from operations. These people are less trained in using scientific approaches when solving problems, finding bottlenecks and taking decisions: they rather depend on observations in the operations. Although this leads to good results in many cases, the risk of taking the wrong solution, or a solution that is less effective than expected, is relatively large. Moreover, when it concerns solutions of a novel nature, as is the case in robotized container terminals, experience from the past is not always the best advisor.

Is the success of the simulation approach throughout the design-engineering process dependent on the people that apply it, or can we conclude that the approach as such appears to be a feasible and value-adding one? As we have applied the approach in the two test cases, as well as several others (see e.g. Dobner et al., 2003a), it is our opinion that this is the way to go to cope with the complexity of the design. Of course, we are aware of our own influence on the outcome, but we also observe the value of the models themselves as integral part of design, development and decision-making. The large amount of guessing that usually takes place, is significantly reduced when the effort to answer it by means of a model is acceptable in terms of lead time and costs.

This brings us to the model system itself. It seems to be ideal to have a model answering questions, but it puts much emphasis on the *validity* of the model. Especially in the case of novel concepts, the availability of empirical data is limited or none. Here the use of validated building blocks (or components) already used in other projects can reduce the questions around the validity. Especially on behalf of
equipment models, this approach appears to be feasible. Secondly, the importance of intelligent and feasible control rules and algorithms is high: it is our experience that a terminal performs according to the intelligence of the control system (in real systems implemented in a Terminal Operating System). Because the TOS for these novel terminals does not exist yet, the documentation of the applied rules within the model system is of eminent importance, and so is the verification of the way they are implemented into the TOS in a later stage.

With regard to validity of models, we also have experienced that trust in models is an important aspect of validity. Trust is a complicated notion, as it does not only depend on the quality of the model, but certainly also on the trust in the modeller and the way he interacts with the clients of the model. During the process, the client should get familiar with the model, long before the results are presented. Otherwise, the results are only trusted when they are exactly what the client expected: as soon as they are deviating from his expectation, he will immediately cast doubt on them. Therefore, the interaction with the client cannot start early enough.

Besides the general observations with regard to the approach for designing automated terminals, we summarise our findings regarding the other guidelines in the next section.

9.4.2 Experiences with regard to the guidelines

In the two test cases, the concrete experiences with most of the guidelines as proposed in the chapters 3 to 6 have been discussed. Here, we mention a number of more generic findings that we gathered during our research.

Although none of the projects we carried out during our research is the same – it comprises new developments, terminal extensions, and terminal improvement programmes - the approach proved applicable to all of these different types. The scope and the set of feasible solutions are determined by the type of project, but in terms of analysis methodology, we conclude that the simulation approach from start to finish appears to be viable. Even in projects where product-oriented TOS software is being implemented, many questions have to be answered, and many uncertainties concerning performance and technical robustness remain where the application of simulation as a test environment early in the process increases the
focus on performance and enables testing that is in traditional approaches only possible after going into operation.

A crucial contribution of the simulation approach in the context of an entire design-engineering project appears to be the see-throughs that have to be made, already when creating the functional design. The see-throughs consist of first detailed designs of solutions that usually only exist in words, but which have to be detailed in order to implement it as a prototype in a model system. This process of elaboration provides not only feedback to the functional design; it also provides an outlook to the technical design in terms of a prototype. Especially when the prototype is built with this additional purpose in mind, i.e. in accordance with the architecture of the entire system, it can provide useful input to the technical specification and the implementation.

This prototyping functionality by means of simulation models is of high added value because most terminals are similar, which allows for re-use of components in the model suite. Due to local circumstances – physical shape of the land, location of rail connection, type of container flow, et cetera – the rules within the TOS have to be configured and refined in every case. In the case that tailor-made software is built or rebuilt, which is quite often so, the design activities comprise more than just configuration. Here, the prototyping functionality provides even more value.

The use of a simulation approach allows for continuous attention for performance issues on top of than getting the system to work. As every project is under time pressure, it is common that during implementation/realisation the focus of the people involved – including the managerial level – moves from the functional requirements to the basic requirement to get the terminal running. The attention for performance issues is then deliberately delayed to a later point in time. However, this may cause it to be impossible changing things at this later point in time, because of decisions made to bring the system to work – as clearly can be seen from the ECT-DSL case (see section 2.7). When a model system is available allowing the people involved to assess whether these decisions would affect performance, different solutions can be searched and decision can be carried through knowing the consequences. Of course, it is essential to make sure that this environment is valid for these kinds of assessments.
In addition to the previous point, the simulation approach can not only be used to keep the focus on performance besides getting the system to work, but it also can be used to test the system under all kinds of circumstances. This means that the likeliness of huge performance losses in the case of break-down, or other disrupting factors, becomes less, herewith reducing the risk for the terminal operator. When the experiences of these tests are shared with the people that will operate the terminal after going live, the negative effects are likely to be smaller during the operation.

A prerequisite to success of the approach proposed in this research is the cooperation and coordination between the simulation group and the software engineering team. We have experienced that this process is not always easy because of two reasons. The main reason is the difference in objective of the development of a specific component. The prototype in the simulation environment is built to assess the contribution to performance, rather than to develop a piece of software that can be used in an operational environment. Although the two can go together, this is not always the case, which may lead to a solution that is perfectly working in a simulation environment, but difficult to transfer to the production software. The second reason is the concurrency between the simulation approach and the software engineering. As the simulation approach continues during the software development, the detailed design is running on two tracks that may deviate at some points. It requires regular exchange of ideas and designs to keep it on the same track.

9.4.3 Contribution to the field of research

The approach we propose and have applied can be characterised by the longitudinal use of models: throughout the design-engineering process we use the same model suite for different purposes. The benefit is reduced development time, the ability to use validated model components, and continuity of information. With regard to the latter, we have experienced that during a process the team of people involved changes, which inevitably leads to a loss of knowledge about the project history. Any way of capturing this knowledge is in our opinion worthwhile because decisions are taken for a reason. The continuity of models enables capturing information, and allows for rewinding: simply to check again why a certain decision was taken. This longitudinal use of models strongly depends on the building-block based way of modelling. Because components of varying aggregation level (and therefore) content, but with the equal interfaces remain inside the building block library, the knowledge
and information behind these also remains and can be re-used whenever opportune. As simulation is a time-consuming approach, every attempt should be made to shorter the model development time at the benefit of experimentation and assessment of alternatives: the longitudinal use of models certainly contributes to this point.

Simulation is a well-known term in software engineering. However, in many cases, the way simulation is used differs from our approach in this research. In many projects equipment simulators are used to represent equipment or machines that are not yet available at the time of software development. However, in most cases, these simulators are finite-state machines without too much dynamic behaviour implemented. Besides, when the system environment allows for interaction between these simulators, in most cases it is not considered: the next status simply occurs after a predefined time, not considering the state of any other machine or piece of equipment. Our proposed approach explicitly considers the interactions between the physically present objects, such as cranes and transport vehicles. Especially these interactions make a system complex and difficult to control, which makes a test-environment containing these elements so valuable. In our opinion, these experiences provide useful insight for the software engineering world, and enrich the set of test environments for complex systems.

9.5 Future outlook and further research

In our approach, we propose applying a simulation approach in all steps of the design process, including the monitoring after the terminal has become operational. Apart from a monitoring function, a simulation approach may also be used to support real-time decisions. For instance, when something unexpected happens, the actual situation may be loaded into the model system, with which then multiple courses of actions can be analysed. The outcome – i.e. the best course of action - may then be fed back into the real system. Although this is principally a typical simulation cycle, the requirements to the model system are of such a nature that this kind of use is not trivial yet. First, it requires an on-line interface to the actual data sources, as well as it requires a short duration of the experiment. At present, the models that could provide valuable decision support are still too slow; to get solid outcomes at least 8 to 24 hours of runtime is required. We expect that these kinds of
applications will be used more in the future as technology will be more able to support them.

A second application which is in line with the previous one is the use of what we call reverse simulation. With this notion we mean that the past is replayed for analysis purposes. For instance, from an operation of two weeks all events are logged. The log is fed into the simulation environment and replayed so that all peculiar things can be analysed off-line. Then, based on the analysis resulting from the replay, parameters can be adjusted and the effect can be analysed by repeating the replay. This approach, using the same environment as for other problem solving cycles, will lead to the ability to configure a terminal operating system avoiding a trial-and-error approach, which inevitably leads to performance loss in some cases. As the comparison is based on (1) real data, and (2) always is of a comparative nature – after all, first the settings that are logged are replayed, then the new settings are analysed – problems with limited expression power are avoided to a large extent.

Finally, we think that the simulation approach for designing container terminals is also applicable for other environments of the same nature, i.e. a high degree of dynamics, a high complexity of tasks, and a high degree of interlacing operations. As example we mention the design of automated baggage handling systems at airports (see for instance Rengelink and Saanen, 2003). Here we also applied a similar approach during design and realisation of a baggage handling system, re-using the models that were used for decision support during the design, for supporting the implementation of the process control systems. A similar approach can be used at a wide variety of systems, factories, distribution systems, warehouses, and so forth.
10 APPENDICES

10.1 References

Agerschou, H., Lundgren, H., Sorensen, T., Ernst, T. et al. (1983), Planning and design of ports and maritime terminals, John Wiley and Sons Ltd.


Bouckaert S. (2002), *Automated Stacking Crane at Hessenatie*, Terminal Operator
An Approach for Designing Robotized Marine container terminals


Chiu H. (2002), Personal communication with people from Hutchinson Port Holding.


Connekt (2002), FAMAS.MV2, Simulatie voor PCS Nieuw, Delft.


Dobner M. and Y.A. Saanen (2001d), presentation at the Port Maritime Association, Oakland.


Dobner M., (2000b), Presentation on developments in container terminal planning and transport, Gottwald Port Technology.


of Felixstowe, internal report, Gottwald Port Technology, Düsseldorf.


Edmond, E.D., Maggs, R.P. (1978), How useful are queue models in port investment decisions for container berths?, *Journal of the Operational Research Society*, 29(8), 741-750.


Fielding

Fishwick P.A., (1992), *An Integrated Approach to System Modelling using a*


Hook van D.J., S.J. Rak, J.O. Calvin, Approaches to relevance filtering, in: *Proceeding of the 11th Workshop on Standards for the Interoperability of Distributed Simulations*.


An Approach for Designing Robotized Marine container terminals


Larkin J.H., H.A. Simon (1987), Why a diagram is (sometimes) worth a thousand


McCubbrey D.J. (1999), Disintermediation and reintermediation in the U.S. air travel distribution industry: a Delphi study, in: *Communications of the Association for Information Systems*, Volume 1, article 18, University of Denver.


An Approach for Designing Robotized Marine container terminals

Internet: 29/01/2001.


Rittel H.W. and W.M. Webber (1972), *Dilemmas in a general theory of planning*,
Working paper 194, Berkeley: University of California.


Saanen Y.A. (2001b), Personal communication with different shipping lines and port authorities (Port Maritime Association, P&O Ports, Port of New York).

Saanen Y.A. (2002a), Personal communication with terminal operators (ECT, Eurogate, Maersk-Sealand).


1-8.


Uglyl, L. (2001), personal communication.


Velsink, H. (1997), Ports and terminals: planning and functional design (CTwa4330), Delft University of Technology.


for the Research program on Building blocks for Effective Telematics Application Development and Evaluation, Delft University of Technology.


Watanabe, I., *Container Terminal Planning, a theoretical approach*, World Cargo News Publishing.


Westland, J.C. & Clark, Th.H.K. (1999), Global Electronic Commerce: Theory and
An Approach for Designing Robotized Marine container terminals

Case Studies, MIT Press.


Ypma, 1999, Towards total terminal service: Development of an AGV- fleet management system (FMS) for a container terminal, master thesis, Delft University of Technology.


10.2 Interviews Delta Sealand terminal ECT (1982-1992)

10.2.1 Introduction
In the beginning of 2002, we carried out 5 interviews with key people that were involved in the design and realisation of the Delta terminals of ECT. The people interviewed had received the questions in advance, together with the following introduction:

The purpose of this interview was to depict the design and implementation processes of the ECT Delta terminals in which the focus lay on the design and implementation of the operating system. Using the design and implementation processes from a number of recent cases as a basis for comparison, an approach was attempted in which simulation was employed from the first analyses to the implementation stage and if possible during the operational process as well.

The minutes of each interview are reported in this appendix. The people interviewed have read and checked the minutes of the interviews afterwards (in Dutch).

10.2.2 ECT PCS 1 Interview with Ron de Waard on 20 December 2001

Questions Concerning the Design and Implementation Processes

Can you state what your role was during the design and realization processes?

In the prototyping trajectory I was involved as one of the lead engineers in the implementation process of an AGV-ASC interface and also in the development of the PAS section.

Can you indicate in which phases the design and realization processes were carried out? In which of these phases were you yourself involved?

<table>
<thead>
<tr>
<th>Definition phase</th>
<th>1989</th>
<th>Not involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototyping: Is it realizable?</td>
<td>1990</td>
<td>Involved</td>
</tr>
<tr>
<td>Basic design (functional)</td>
<td>1990-1991</td>
<td>Not involved</td>
</tr>
<tr>
<td>Implementation</td>
<td>1991-1994</td>
<td>Involved</td>
</tr>
<tr>
<td>Testing phase</td>
<td>Late 1992</td>
<td>Involved</td>
</tr>
<tr>
<td>Start-up phase</td>
<td>1993</td>
<td>Involved</td>
</tr>
<tr>
<td>Fine tuning</td>
<td>1994</td>
<td>Involved</td>
</tr>
</tbody>
</table>

Was each of those phases closed with clearly documented decisions? How so?
For a long time, the functional design process was carried out simultaneously with the implementation process. In other words, a rough architectural plan existed from the very start, and the interfaces were well established. However, the functions were filled out during implementation. So from the start of the implementation process, there already was extensive documentation with regard to the components that were already completely designed.

Was there a clear relationship between how the decisions were made based on what had occurred in the previous phase?

N/A

- **Questions Concerning the Parties Involved and Their Role in the Design Process**

*Can you indicate what the key players’ roles were during each phase and the decision-making process?*

The development of the operating system was subdivided into two separate sections: PAS and PCS. Project managers were appointed for both sections, Rob Lijn for PAS and Rene Besselink for PCS, respectively. Said sections functioned primarily on their own based on the defined interfaces. In other words, they had little to do with one another. These two design/development sections were then further subdivided into design teams. Each team had an active designer who was responsible for developing the blank spots. In addition, there were at least three people from the operational process who functioned as a sounding board for the entire development section.

*Do you believe that all of the relevant parties were involved in every phase (keeping in mind that the relevant parties might differ per phase)?*

N/A

*Can you expand on the role of the people involved with regards to the disciplines and the “division of power” during each phase of the design and implementation processes?*

There was a heated rivalry between the PAS and PCS subsections based on which sections were responsible for which specific developments. The relationship between PAS and PCS was further strained by the enormous amount of pressure top get things done on time.
Questions Concerning the Subject Matter of Each Phase in the Design and Implementation Process

Was it perfectly clear in every phase as to:

- What the purpose and/or design task was?
- Which boundary conditions existed due to previously-made decisions?
- What support information was available?
- Who the client and/or interested parties were?
- What requirements were placed on the design and implementation processes?

The interfaces were the most important boundary conditions (“the interfaces were sacred”). As to performance (both technical and functional) the only requirement was that “it has to work.” There was extensive documentation of what had been designed, functionally as well as technically.

One of the boundary conditions (or basic assumptions) that had an important impact on the project was that the equipment was infallible. This in turn resulted in software that was only marginally suited for catching equipment mistakes.

Implementation was first directed at the core elements of the PAS and PCS sections, and later additions such as MTs and ITT were built in.

Can you state what role simulation had in each of the phases?

What results were achieved?

What were the advantages and disadvantages of using simulation?

Much of the software was tested as a module, e.g. by testing it with an equipment clone (“simulation”). Additionally, software-integration tests were performed by again utilizing clones. This would have been a very successful manner of testing had it not been for the assumption that the equipment was infallible. This led to limits on the testing. The program was unable to compare the functional quality with the technical performance of the software. Instead, the tests were aimed at getting the software up and running. Besides this, the pressure to get everything done on time was such that much of the sophistication of the implementation process was dropped in favour of simpler solutions. Simulation of the entire system coupled with finished software components did not take place during the implementation process.
Can you indicate how “implementable” the functional requirements were when the development of the operating system began?

A certain level of functionality was not realized during the implementation process for several reasons. For one, because blank spots still existed for that level of functionality. For another, the pressure to get things done on time played an important factor in choosing simpler solutions.

Can you indicate in what way an effort was made to realize the functional requirements and test the performance of said requirements during the implementation process? In your opinion, which role would simulation have played in this?

During implementation and the various testing phases, functional performance was given a low priority. The goal of the tests was to get the system up and running.

What do you believe to be the three most important reasons as to why the system did not function as planned and specifically in regards to simulation?

The complexity of the system was underestimated.

The assumption that the system was infallible led to several problems during testing and in practice, especially since problem-solving techniques had not been predicted to be necessary.

In some cases, compromises instead of rational assessments determined how components were interfaced.

10.2.3 ECT PCS 2 Interview with Peter Metsch on 25 January 2002

- Questions Concerning the Design and Implementation Processes

Can you state what your role was during the design and realization processes?

I had a number of different roles as: 1. An implementor (of the AGV-ASC interface) in the prototyping phase; and 2. A designer/implementor during the building phase of PCS (Order Control System, OCS), after which I was involved in the improvement trajectory for PCS (until 1999):

- Changeover from simulation specifications to implementation reports.
- Total integration of the OCS framework design.
- Design, team leadership, and development during the realization process.
- Improvements during testing and operation processes.
An Approach for Designing Robotized Marine container terminals

- Clearing up omissions, for example loading process, topology, transitions and Seaside – Landside cooperation.

Can you indicate in which phases the design and realization processes were carried out? In which of these phases were you yourself involved?

<table>
<thead>
<tr>
<th>Phase</th>
<th>Start/End</th>
<th>Involvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition phase</td>
<td>1989</td>
<td>Not involved</td>
</tr>
<tr>
<td>Prototyping, is it realizable</td>
<td>1990</td>
<td>Involved</td>
</tr>
<tr>
<td>Basic design (functional)</td>
<td>1990-1991</td>
<td>Involved</td>
</tr>
<tr>
<td>Implementation</td>
<td>1991-1994</td>
<td>Involved</td>
</tr>
<tr>
<td>Testing phase</td>
<td>End of 1992</td>
<td>Involved</td>
</tr>
<tr>
<td>Start up phase</td>
<td>1993</td>
<td>Involved</td>
</tr>
<tr>
<td>Fine tuning</td>
<td>1994</td>
<td>Involved</td>
</tr>
<tr>
<td>Re-design projects</td>
<td>Up to 1999</td>
<td>Involved</td>
</tr>
<tr>
<td>PCS-Nieuw</td>
<td>From 2001 onwards</td>
<td>Involved in specifications for PCS-Nieuw</td>
</tr>
</tbody>
</table>

I was not involved in the feasibility process (definition phase). As part of information, we received algorithm specifications that had been developed by the TOR (Technology and Operation Research) group and they had been categorized as being sufficient. Unfortunately, whether they were sufficient could not be verified by anyone outside of the TOR group itself.

Period from 1989 to 1996.

In 1996, the “terminalbesturing” project was started as a subsection of 2000-8. At that time, PCS developments were stopped. When “terminalbesturing” was also stopped at the end of 1998, I made some modifications in the existing PCS code based on the new design for “terminalbesturing” (until March, 1999). In late 2000 to early 2001, I was involved in the PCS-Nieuw specifications (please note, these did not describe new required functionalities or specs, but primarily described the implemented functionality in PCS (for Dynacore)).

Was each of those phases closed with clearly documented decisions? How so?

For a long time, functional design and implementation were carried out simultaneously. From the start of the implementation phase, there was a rough architectural plan (the OCS framework), and the interfaces were well defined, up to the point that changes were not made/carried out. The communication between the designers (TOR group) of the OSC algorithms and the implementers was poor and not specific enough.
At the start of implementation only a description of how functionalities should be operationalised (for instance, the routing of orders to AGVs as well as the stacking algorithm) was available. The members of the implementation team did not have any access to further information such as pseudo code nor did they have the possibility to verify implemented code using simulation models.

*Was there a clear relationship between how the decisions were made based on what had occurred in the previous phase?*

The project cannot be categorized as having phases since the design (both functional and technical) largely took place simultaneously. This occurred in part because there was not even a design during the implementation phase that was suitable for implementation.

Additionally, there was barely an integrated design process, seeing as how a large amount of decisions had already been made (e.g. the layout) in spite of the fact that the distance to the quay stack was too limited for a decent traffic flow.

These remarks apply mainly in regards to OCS (the most important subsection of PCS). Other PCS subsections did have a phased project trajectory. The problem in OCS was the deadline on the one hand and that the TOR group took too long to deliver the needed specifications on the other. Later, said specifications proved to be insufficient.

- **Questions Concerning the Parties Involved and Their Role in the Design Process**

  *Can you indicate what the key players’ roles were during each phase and the decision-making process?*

The OCS algorithm design was done by the TOR group. This group delivered descriptions of the algorithms (in place of code), after said algorithms had been tested using simulation. The algorithms were implemented and then tested on how they functioned technically. Again, the interaction between the design and implementation groups was marginal at best. Because some of the algorithms were not finished on time (e.g. routing), the implementation group had to implement without having an actual design. Besides the basic implementation group, there was also a back-up group that implemented an alternative variant. Said alternatives were even less sophisticated than the standard implementations.
There was an independent quality-control group that tried to evaluate the designs, but with respect to the performance and algorithms, they had to trust the TOR group. In addition, there was an independent testing team that attempted to test everything as much as possible before it went into production.

Do you believe that all of the relevant parties were involved in every phase (keeping in mind that the relevant parties might differ per phase)?

N/A

Can you expand on the role of the people involved with regards to the disciplines and the “division of power” during each phase of the design and implementation processes?

N/A

- Questions Concerning the Subject Matter of Each Phase in the Design and Implementation Process

Was it perfectly clear in every phase as to:

- What the purpose and/or design task was?
- Which boundary conditions existed due to previously-made decisions?
- What support information was available?
- Who the client and/or interested parties were?
- What requirements were placed on the design and implementation processes?

The only role that performance (technical as well as functional) played during the implementation process was “it had to work”. From that which had previously been designed, there was extensive documentation available, however, not in a way so that it could only be implemented in a single way. Moreover, the implemented functionality often proved dysfunctional.

One example of this was the order assignment module that made for an unusually uneven servicing of the loading and discharging cranes. Moreover, much of the specified functionality had not been developed completely; for instance, the parameters for stack organization were missing completely.

Furthermore, the entire design had not been set up in terms of a total system concept; for instance, the sea-/landside interaction only functioned whenever the
equipment (quay crane/AGV/ASC/Carrier) did not break down. However, if there were any disturbances (breakdowns, or lunch breaks), landside equipment was given a disproportionately high priority.

In spite of the implementation group’s perception that the TOR group had a complete simulation model, this later proved to be untrue. This led to many design concepts being tested by models that modelled part of the system, through which some parts of the system dynamics were poorly designed in terms of functionality.

*Can you state what role simulation had in each of the phases?*

*What results were achieved?*

*What were the advantages and disadvantages of using simulation?*

The TOR group used simulation as a tool to test certain principles. A disadvantage to this approach was that they did not have a complete model to portray each and every process. The first implementation (late 1992 continuing into 1993) was partly built based on the algorithm specified by the TOR group and partly self designed algorithms (for lack of anything better).

*Can you indicate how “implementable” the functional requirements were when the development of the operating system began?*

In general, it can be said that there was a communications gap between the TOR and implementation groups. In addition, the project deadline caused additional stress since the financing department demanded a first operation of handling vessels before the end of 1992. In fact during this first operation, the system was in no way operational.

Furthermore, the algorithm that was handed in, was in an insufficient format (descriptions in place of code). It is quite possible that this led to many of the discrepancies between the design and implementation groups, although the designs often proved to be dysfunctional in practice.

*Can you indicate in what way an effort was made to realize the functional requirements and test the performance of said requirements during the implementation process? In your opinion, which role would simulation have played in this?*

Functional performance was given a low priority during implementation and in various test phases. Tests were aimed at getting the system up and running.
Simulation was partly used in designing the system, but the entire system has never been analyzed. The performance requirements as defined (260 moves per hour for a complete operation) were based on simple calculations in terms of the capacity of delivery lanes to the quay cranes. These calculations brought about the assumption that the lanes could completely be used.

Uncertainty exists concerning the relationship between the AGV throughput capacity in the delivery lanes and the realizable production in the number of containers by the quay crane. This last quantity is influenced by more than just the routing capacity alone.

Once again, simulation was barely used in the improved trajectory. Instead, the PEP environment (the emulation of actual software, YAS) was used to test improvements.

*What do you believe to be the three most important reasons as to why the system did not function as planned and specifically in regards to simulation?*

Expectations were far too high. Even using today’s technology and systems, the goal is not realizable!

The technology (for both hard- and software) came up short.

The necessary interaction between manual and automated operations had been underestimated. Actually, this has yet to be optimized. Due to misunderstandings and ill will, the performance suffered.

There were culture clashes between the operators and the supporting staff, which led to a poor working relationship.

- Additional remarks

The OCS architecture was data oriented instead of process oriented.

The architects had little or no knowledge of how an automated terminal should function, though they did have knowledge of the container business (e.g. reefer facilities, off-standard, IMO, MTs, etc.)

Had the TOR group and implementation team been integrated, several of the existing problems would have been averted.

The container terminal and the ASC-stack in particular were used differently than originally planned: The stack was used more like a European storage area by Sea-land
than originally foreseen. This quickly led to the stack capacity being expanded and
more and more was stacked. Even the concept of Majority Sea to Sea (modal split)
was different in practice than predicted.

In the algorithms, operational circumstances such as stack occupancy, berth
schedules, (arrival schedules for vessels) quay crane production, landside operations,
and the arrival and departure of vessels (which had consequences for the deployment
of the quay cranes) were barely taken into account.

Technical/operational disruptions were not taken into consideration in the
algorithms: crane stoppages, broken down AGVs and ASCs, absent crew/personnel
of ASC/Ms or FLTs.

10.2.4 ECT Results Interview with Hans Veeke on 9 February 2002

- Questions Concerning the Design and Implementation Processes

*Can you state what your role was during the design and realization processes?*

As a member of the Technical and Operations Research (TOR) group (five members
strong), I was involved, in the functional specifications trajectory from 1987 to 1991,
and then from 1991 onwards, I was part of the improvements trajectory. My primary
responsibility was the AGV system. From the beginning in 1987, the layout and
handling systems (carrier, ASC, AGV, QC) had clearly been defined. Questions as to
the design of the various sections (landside, stack, horizontal transport, and quay
cranes) were divided among the members of TOR. A detailed, holistic design was
virtually nonexistent.

From the start of the AGV system’s design phase, the focus lay mainly on basic
principles given that an AGV system of this size had never been (and to this day has
yet to be) depicted in such detail. During the design phase the scope had been
restricted to the AGV system. The same is true for the simulation models.

*Can you indicate in which phases the design and realization processes were carried out? In which of
these phases were you yourself involved?*

<table>
<thead>
<tr>
<th>Phase</th>
<th>Years</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploration phase</td>
<td>1982-1987</td>
<td>Not involved</td>
</tr>
<tr>
<td>Definition phase</td>
<td>1987-1990</td>
<td>Involved as TOR member</td>
</tr>
<tr>
<td>Prototyping: Is it realizable?</td>
<td>1990</td>
<td>Involved</td>
</tr>
<tr>
<td>Basic design (functional)</td>
<td>1990-1991</td>
<td>Involved</td>
</tr>
</tbody>
</table>
Was each of those phases closed with clearly documented decisions? How so?

The deadline for the delivery of the functional design was August 1991. At that time, a document was handed in describing a large part of the functionality. However, not all of the specifications provided solutions that were implementable. For instance, one of the specifications dealt with AGV malfunctions: The other AGVs were supposed to circumvent a broken-down AGV. However, exactly how the malfunctioning AGV should be detected, or which alternative routes should be available were never specified in any detail. Because of this, the functional specification phase continued during implementation.

The functional specifications were delivered in English. These were based on an algorithm that had been developed and partly tested in a simulation environment. Using these specifications as a basis, the implementation section had the job of developing the software. However, there was no pseudo-code or the like in the English specifications, which in turn led to distinct differences between the functional design and the implementation thereof.

Was there a clear relationship between how the decisions were made based on what had occurred in the previous phase?

Initially, the project should have taken place in clearly defined linear phases. Theoretically, each phase should have been closed before moving on to the next one. In practice, this approach did not work. Had the implementation section gotten involved earlier, as well as defining a couple of process iterations, things might have turned out for the better. In addition, the pressure to get everything done on time had an impact on the project. Besides, there was hardly any integration whatsoever during the design process, since a great deal was already in place (e.g. the layout and choice of handling system), in spite of the fact that the distance from the quay to the
Questions Concerning the Parties Involved and Their Role in the Design Process

Can you indicate what the key players’ roles were during each phase and the decision-making process?

N/A

Do you believe that all of the relevant parties were involved in every phase (keeping in mind that the relevant parties might differ per phase)?

It would have been nice if the functional design and implementation sections had worked more closely together.

Can you expand on the role of the people involved with regards to the disciplines and the “division of power” during each phase of the design and implementation processes?

N/A

Questions Concerning the Subject Matter of Each Phase in the Design and Implementation Process

Was it perfectly clear in every phase as to:

- What the purpose and/or design task was?
- Which boundary conditions existed due to previously-made decisions?
- What support information was available?
- Who the client and/or interested parties were?
- What requirements were placed on the design and implementation processes?

First of all, the goal was to create a system that had never been designed before. The project was therefore a difficult task, since the amount of time allotted was also limited.

Additionally, from the start of the functional design (in 1987), several restricting boundary conditions were set on the handling system’s layout and dimensions (48 operational AGVs). Consequently, functional questions that still needed to be
answered were limited by the boundary conditions. In short, the freedom to design was severely restricted.

The basic set of principles was partly derived from experiences with conventional terminals. Of these, the two most important that may be called into question, were: 1. A quay crane should never be kept waiting and 2. Once a container has been stacked, it should never be relocated. The first because of the unlikelihood to realize it and the second, because in an automated terminal the relocation costs are minimal and relocation can clearly improve the performance of the terminal. A better objective for the quay cranes would have been to regulate the arrival times of the AGVs (to every 60 seconds, for example).

In many areas, the information that the TOR group possessed was incomplete. Thus the behaviour of the quay cranes (specifically large variance of the cycle times) as well as the AGV arrival procedure at the cranes was too unpredictable, which caused the actual process to significantly differ from the simulation results. This resulted (in conjunction with these reasons) in traffic jams and ensuing blockages under the quay cranes and decreased efficiency in dealing with the AGVs.

As to the technical layout (algorithm), the design may be termed as a greenfield situation; there were few, if any, precedents (and certainly not in comparable situations) to use as an example. For instance, the claiming of AGVs, routing, topology, and order allocation had to be completely developed from scratch. Coupled with the increasing pressure to get everything done on time, the goal of the project went from having a good performance to simply having a system that worked. This, in turn, negatively affected the sophistication level of the project.

*Can you state what role simulation had in each of the phases?*

*What results were achieved?*

*What were the advantages and disadvantages of using simulation?*

The TOR group used simulation to test specific principles. Simulation models that represented parts of the system, were designed and turned out to be limited and insufficient, making it impossible to test the system as a whole, as desired. One important reason for this was the limited capabilities of the computers themselves. In theory, it should have been possible to use these simulation models to test parts of the system.
However in regards to interfaces (interaction between the various system parts), there was not enough support by simulation, especially due to technical reasons. While in some areas (including claiming), the simulation environment was quite detailed, the manner in which the transition to implementation was realized (describing the code instead of pseudo-code) led to implementation that deviated from the specifications. In addition, many problems, such as how to deal with breakdowns, had not been adequately anticipated.

Can you indicate how “implementable” the functional requirements were when the development of the operating system began?

The TOR and implementation groups did not work well together. Only at a much later stage (during 1993-1995) did they try to pull together to put improvements into practice.

Furthermore, the algorithms given to the implementation group were unwieldy and open to multiple interpretations (descriptive texts). This might have led to discrepancies between design and the implementation. In addition, not everything had been properly specified, so the implementation group improvised without testing the effects on performance while implementing.

Can you indicate in what way an effort was made to realize the functional requirements and test the performance of said requirements during the implementation process? In your opinion, which role would simulation have played in this?

Functional performance was not a high priority during implementation and the various testing phases. Even in the Start-up phase before the operation got underway, the TOR group was not optimisation-oriented, but more concerned with getting the system up and running.

What do you believe to be the three most important reasons as to why the system did not function as planned and specifically in regards to simulation?

- A shortage of time (capacity), in relation to the scope of the project.
- Inexperience in designing a completely new, unprecedented, automated terminal system.
- The division between the disciplines and how things were done was too great.
**Additional remarks:**
The architecture of the system was not designed from a holistic viewpoint.

In practice, technical causes (e.g. overloading the computers that ran the operating system: whenever other programs were run on these computers, the performance level of the operation decreased) also proved to be among the reasons that the system did not perform according to expectations.

### 10.2.5 ECT PCS 3 Interview with Rene Besselink on 18 February 2002

#### Questions Concerning the Design and Implementation Processes

*Can you state what your role was during the design and realization processes?*

- Project manager for the entire PCS development process.
- Start-up manager for the introduction at the terminal.
- Project manager for setting up the software maintenance.
- Involved at PCS from 1987 until 1995.

*Can you indicate in which phases the design and realization processes were carried out? In which of these phases were you yourself involved?*

- Global basic design controls; splitting the coordination module (later OCS) and equipment controls (later ASC and AGV control).
- Global basic design of technical possibilities and the development direction; rough functionality sketch; deciding where (at the coordination and equipment levels) functionality should be implemented.
- Splitting up control units: 1. Total coordination of units (OCS); 2. Steering and controls for each type of equipment; 3. Monitoring and reporting (e.g. amount of operating hours); 4. Testing facilities.
- Realization for every control unit: 1. Functional design; 2. Technical design; 3. Realization; 4. Integration tests.
- Implementation: 1. Pilotplant; 2. Starting-up the terminal.

*Was each of those phases closed with clearly documented decisions? How so?*

Yes, during each phase and section, the project group (made up of the project manager and project leaders, and depending on the decision, helped by mechanical engineering, simulations, operations, etc.) made a Go/No Go decision.
However, a Go/No Go decision did not mean that a specific phase had actually been ended. In practice the functional and technical design were carried out simultaneously as implementation due to the continuous engineering. Another result was that a number of functions (AGV control, for example) were significantly revised a number of times when the implementation took place. Furthermore a lot of aspects (e.g. dynamic routing) never saw a finished version of its functional design.

Was there a clear relationship between how the decisions were made based on what had occurred in the previous phase?

Yes. The decisions that were made had an enormous impact on subsequent phases as well as the preceding phases of other sections.

- Questions Concerning the Parties Involved and Their Role in the Design Process

Can you indicate what the key players’ roles were during each phase and the decision-making process?

Mechanical engineering, simulation, and civil engineering were the “key disciplines”. These disciplines mainly looked at the results of a phase and judged the results based on their expectations. This led to several revisions due to the progressive and ever-changing outlook.

In the pre-phase from 1982-1987, the systems engineering and IT disciplines were only marginally involved. In hindsight that was not such a good idea, since for an automated system, hardware and software designs should be preferably made interactively.

Do you believe that all of the relevant parties were involved in every phase (keeping in mind that the relevant parties might differ per phase)?

No, the most important “party” was the operation itself. Although it did not yet exist, it was represented by the people who would be in charge of operations in the future. Input on a detailed level was only successfully carried out during implementation. Furthermore, it was especially difficult for those involved in operations, if not impossible, to visualize the new system. One of the consequences of this was that the design was an automated version of a conventional terminal (at least in a number of areas). Had it been possible to better visualize the final product, the quality of the design might have been better.
The entire project was characterized by continuous engineering.

*Can you expand on the role of the people involved with regards to the disciplines and the “division of power” during each phase of the design and implementation processes?*

In building PCS, mechanical engineering and simulation led the way. Gradually, mechanical engineering made way for simulation and operations. Because the functional specifications were not ready at the delivery time, the implementation section enjoyed a great deal of freedom. On the one hand, this was good for the project, on the other hand, implementation was supposed to take place on the basis of the specifications.

- **Questions Concerning the Subject Matter of Each Phase in the Design and Implementation Process**

  *Was it perfectly clear in every phase as to:*

  - What the purpose and/or design task was?
  - Which boundary conditions existed due to previously-made decisions?
  - What support information was available?
  - Who the client and/or interested parties were?
  - What requirements were placed on the design and implementation processes?

No, because there was a constant search for the best possibilities from IT as well as mechanical engineering. Operational decisions were made on the basis of conventional terminal experiences, of which the practicality often could only be tested at a late stage.

The boundary conditions that mechanical engineering made were constantly changing during the project due to the ongoing development of equipment.

At the start of implementation many requirements were still unavailable, because of the time it took to develop simulations as well as the lack of computer power.

*Can you state what role simulation had in each of the phases?*

*What results were achieved?*

*What were the advantages and disadvantages of using simulation?*
During the first phases, simulation was mainly directed at equipment usage. For instance, simulation was used to predict how AGVs would behave when routed. Design choices were made based on these simulations and attempts were made to translate said simulations into software.

The most important benefit that simulation brought to the project was in helping create the functional and technical designs for the controls of the equipment.

The greatest disadvantage was that due to too many variables, it was difficult to compare the software to the simulation. Certainly, this was the biggest problem encountered in testing the software. The simulations were far more limited than the software because not all of the software’s capabilities had been programmed into the simulation. Moreover the software needed more variables due to practical infeasibilities.

Another problem was that the simulations only modelled parts of the system instead of the system as a whole. This made the ensuing interactions between the modelled parts difficult to predict.

In addition, the TOR group developed a number of “unimplementable” concepts; sometimes this was due to the fact that the algorithm was not designed in a system environment and sometimes because the calculations took too long and would not have worked in a real-time system.

*Can you indicate how “implementable” the functional requirements were when the development of the operating system began?*

That’s very difficult. There was definitely a sort of broad outline as to how things should be done, but the requirements themselves kept changing. The requisite functionality was still wanting. That led to “shooting at moving targets.”

*Can you indicate in what way an effort was made to realize the functional requirements and test the performance of said requirements during the implementation process? In your opinion, which role would simulation have played in this?*

One of the biggest problems in creating the software was how it should be tested. It was difficult to set up a completely representative test environment for the software. The equipment had to be simulated, and this was done by creating special testing software. Simulation became the norm that had to be met. After testing simulation
was used to identify functional improvements, that could be realized in terms of performance and implementation.

However, the software development was specifically directed toward finding reliable solutions. So that led to various (simpler) back-up solutions being implemented in order to at least have something which functioned.

What do you believe to be the three most important reasons as to why the system did not function as planned and specifically in regards to simulation?

The simulations were divided into subsections, mostly equipment oriented. Therefore, it was difficult to determine what the ensuing effects were on integration.

It was not possible to take all variables into consideration in the simulations.

The required performance level could not be met using the defined algorithms in the production software, often due to limited computer performance.

- Additional remarks:

In an automated system, the people involved in setting up the operation must be trained differently than those who work at conventional terminals. Additionally, thorough training must be given to those who need to operate the system.

During the development phase, most of the decisions (e.g. as to where a container should be stacked or which route an AGV should follow) were made by the users, who thought in terms of a conventional terminal. In the end, only some of the users’ decisions were included in the software, and even these proved to be too many!

Not enough attention had been spared for dealing with breakdowns. Malfunctions were assumed to be exceptional, whereas in practice this assumption proved to be woefully wrong. Even in the simulations, little attention was paid to breakdowns.

The development done by PAS and PCS, which was for the most part done separately, was seen as feasible. The working relationship between the two groups was not optimal.

10.2.6 ECT PCS 6 Interview with Ruud van der Ham on 21 February 2002

- Questions Concerning the Design and Implementation Processes

Can you state what your role was during the design and realization processes?

- Involved in the pre-design of the handling system (from 1984 onwards).
- From the start-up of the terminal, involved in modernization projects such as 2000-8 and in terminal operations.

*Can you indicate in which phases the design and realization processes were carried out? In which of these phases were you yourself involved?*

- Involved in the overall basic design of the control system (1984-1989).
- Involved as the manager of the TOR group in the functional design of the controls system/ equipment controls/ equipment.
- Indirectly involved in the implementation phase.

*Was each of those phases closed with clearly documented decisions? How so?*

Before the actual trajectory of the Delta Sealand terminal had been started, there was a comprehensive pre-phase in which a number of handling system variants were tested using simulation. Earlier, an estimation as to the length of the quay, height of the stack, and transhipment capacity had been made. Comparing this first version of the handling system with the later version, four concepts remained. These four concepts were submitted to an expert’s judgment, which led to how the AGV-ASC system was chosen (1988-1989). In the design phase, IT experts were not involved. Simulations were carried out using the Must simulation package (based on Borland-Pascal) and were suitable for their purpose. The design itself was limited to the hardware (equipment).

The decision to automate had already been made based on an estimation of the costs involved.

The exploration or (pre-design) phase was closed with a distinct preference for a system using AGVs and ASCs, and the layout was for the most part in place. The functional specifications phase was also closed with the delivery of a specifications document.

*Was there a clear relationship between how the decisions were made based on what had occurred in the previous phase?*

Yes, although not all of the decisions are easily recognizable in the resulting system as built.
Questions Concerning the Parties Involved and Their Role in the Design Process

Can you indicate what the key players' roles were during each phase and the decision-making process?

The people and groups responsible for the project’s organization are as follows:
TOR, Ruud van der Ham; PAS, Rob Lijn; PCS, Rene Besselink; ASC, Pieter v. d. Veen; AGV, Frank Nooijen; Labour organization; and Quay cranes.

The TOR group provided support to all of the other groups with a set of simulation models, each modelling part of the system. Also, decisions were first passed to the equipment section and were later handled by IT.

Do you believe that all of the relevant parties were involved in every phase (keeping in mind that the relevant parties might differ per phase)?

In the pre-phase, there were a number of people from the mechanical and civil engineering branches who were involved as well as myself. Unfortunately, these people were not knowledgeable on automation and software engineering.

Similarly, the specifications phase largely lacked input from the software engineering discipline.

During the implementation phase and technical design of the software, software engineering determined the design to a large extent. This led to a great number of changes in regards to the functional design.

Can you expand on the role of the people involved with regards to the disciplines and the “division of power” during each phase of the design and implementation processes?

N/A

Questions Concerning the Subject Matter of Each Phase in the Design and Implementation Process

Was it perfectly clear in every phase as to:

- What the purpose and/or design task was?
- Which boundary conditions existed due to previously-made decisions?
- What support information was available?
• Who the client and/or interested parties were?

• What requirements were placed on the design and implementation processes?

The project was characterized by a great deal of design freedom. Besides, because there was no similar system in existence that could be used for guidance, it was difficult to stay on track.

Can you state what role simulation had in each of the phases?

What results were achieved?

What were the advantages and disadvantages of using simulation?

There were many simulations at diverse abstract levels up until the functional design stage. During functional design, a number of algorithms were prototyped in a simulation environment. One problem with this approach was that developing the models was rather time consuming. Partly on the basis of these simulation experiments, a functional design was created that consisted of a description of the most important components. Reasons as to why the development team chose this method of communication include the readability for non-technical people, the fact that the code in the simulation environment was quite different from that of the production code, and the lack of exception handling in the simulations. The disadvantage of providing information in this manner was that the development team often had difficulty interpreting the document.

Models were only made of parts of the system instead of simulating the system as a whole. An architect for the entire system (for everything, including hard- and software) was missing. Furthermore, an entirely simulated system was impossible to realize at that time due to technical reasons.

Simulation was no longer used from the implementation phase onwards. Also, the fact that the implementation deviated from the functional specifications was no longer evaluated in terms of the total system’s performance. The pressure to get everything done on time played an important role here. In addition, the working relationship between the TOR and implementation groups was quite poor.

Can you indicate how “implementable” the functional requirements were when the development of the operating system began?
It proved to be more complicated than expected to simply transpose the functional specifications into a production code.

*Can you indicate in what way an effort was made to realize the functional requirements and test the performance of said requirements during the implementation process? In your opinion, which role would simulation have played in this?*

It was not.

Simulation could play a role in an implementation trajectory in terms of quality control. Additionally, it could provide more insight into what would happen next.

*What do you believe to be the three most important reasons as to why the system did not function as planned and specifically in regards to simulation?*

Equipment failure.

The unpredictability of the QC cycle.

The operations section not being involved enough during the design process.
10.3 Questions remote survey

The following questions have been asked to a number of simulation or container terminal design experts.

1. I would qualify myself concerning simulation as a:

- User of simulation results or reports based on simulation or other quantitative tools in order to support decision making.
- User of simulation models to analyse systems in the sense of what-if questions and/or scenarios.
- Developer of simulation models, using domain specific components to configure my models.
- Developer of simulation components, focusing on the creation of components with which models can be created.

2. In my daily practice, I use simulation models or tools to:

- Support decision making
- Create more insight into the systems I design or analyse
- Visualise reality
- Develop hardware or software
- Support operational planning
- Improve currently running systems

3. The purpose of simulation is (give points, max 10 per item):

- To visualise and animate the operational process
- To convince my manager/client of the quality of a certain solution
- Analyse systems that cannot be analysed using analytical tools or formulas
- Design equipment
- Test software
- Support day-to-day planning in the operation
- Compare different alternatives
- Prototype control algorithms
- Design an entire system, in this case a container terminal (e.g. terminal shape and size, handling system, logistic control system)
4. A (robotized) container terminal design-engineering project would benefit from the use of simulation as support tool throughout the entire process, so from support of the early idea to final commissioning and operations.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

5. It is a new paradigm in systems development to use simulation as support tool throughout the entire design-engineering process

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

6. Simulation always takes too much time

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

7. A simulation is only useful when an abstract representation of reality (and not too much detail) is applied, because the essence of simulation is to provide a reduced representation of reality.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree
8. The use (and re-use) of validated simulation components reduces the effort of validation of the entire model and contributes to the accreditation of a model:

− Strongly disagree
− Disagree
− Neutral
− Agree
− Strongly agree

9. The use of a scaled 2D animation (what you see is what you get) in the design-engineering process of a container terminal, enables the user to understand the system better than a non-scaled animation does.

− Strongly disagree
− Disagree
− Neutral
− Agree
− Strongly agree

10. The use of a scaled 3D animation (what you see is what you get) in the design-engineering process of a container terminal, enables the user to understand the system better than a scaled 2D animation does.

− Strongly disagree
− Disagree
− Neutral
− Agree
− Strongly agree

11. In the case of an automated terminal, the design of software can benefit from the use of simulation when the architecture of the simulation model equals the architecture of the production software

− Strongly disagree
− Disagree
− Neutral
− Agree
Strongly agree

12. The on-line linkage of production software components (for instance a job assignment component or an AGV router) to a simulation model, contributes to the validation and accreditation of a simulation

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

13. The possibility to test production software components on-line by linking them to a simulation environment representing the entire environment of these software components reduces the time to develop software and improves the quality of the software

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

14. The possibility to switch between abstraction levels (levels of detail) within one model architecture, contributes to the creation of insight and to the applicability of simulation

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

15. In the case of the design-engineering process of a robotized container terminal, it is essential to represent the failure behaviour of equipment in the simulation environment, since it is an intrinsic property of the real system

- Strongly disagree
- Disagree
16. The use of simulation on a detailed level will contribute most to

- The design of equipment (technical specification and kinematic processes)
- The design of software (planning, control, co-ordination)
- The design of equipment and software in interaction with each other

17. Which of the following applications of simulation in the context of a container terminal design-engineering process is beneficial in the sense that it improves/contributes to the design of the container terminal as well as the design process itself:

- Long term planning of quay length, stack capacity, berth utilisation and gate capacity
- Middle-long term planning of resources such as stack, equipment and manning
- Middle-long term assessment of stacking strategies
- Short-term evaluation (peak-operation) and assessment of co-ordination and assignment rules of equipment
- Short-term (peak-operation) determination of the required equipment numbers and specifications (e.g. speed, power, capacity)
- Design of equipment (kinematic behaviour, technical specs) and equipment control algorithms
10.4 Respondents remote survey

<table>
<thead>
<tr>
<th>Name</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbraeck, Alexander</td>
<td>Delft University of Technology, Head of department of Systems Engineering</td>
</tr>
<tr>
<td>Richter, Michael</td>
<td>Innotech Gmbh, principle consultant container terminals</td>
</tr>
<tr>
<td>Ward, Tom</td>
<td>Principal, Terminal Planning &amp; Analysis, JWD Group, Oakland, USA</td>
</tr>
<tr>
<td>Bruzzone, Agostino</td>
<td>Director McLeod Institute of Simulation Science, Genoa Center DIP - University of Genoa Genoa Headquarters</td>
</tr>
<tr>
<td>Franke, Klaus-Peter</td>
<td>Noell Crane Systems, head of system design department</td>
</tr>
<tr>
<td>Van der Ham, Ruud</td>
<td>ECT, former head of logistic consultancy department</td>
</tr>
<tr>
<td>Dobner, Mathias</td>
<td>Gottwald Port Technology, head of system design department</td>
</tr>
<tr>
<td>Uglvig, Laurids</td>
<td>APM Terminals, head of planning department Algeciras container terminal, Spain</td>
</tr>
<tr>
<td>Busk, Kent</td>
<td>APM Terminals, project manager Research &amp; Development, Copenhagen</td>
</tr>
</tbody>
</table>

From two other experts, we did not get a response.
10.5 Samenvatting (in Dutch)

Wereldwijd transport groeit op continue basis. Zelfs tijdens de recessie in het begin van de 21ste eeuw, groeide het intercontinentale transport met 5 tot 10 procent per jaar. Noodzakelijkerwijs diende de afhandelingscapaciteit op container terminals hiermee gelijke tred te houden. Vooralsnog lijkt de trend van grotere schepen en grotere container terminals stand te houden, aangezien de limieten nog niet lijken te zijn bereikt.

Tegelijkertijd neemt de aandacht voor het milieu (ruimtegebruik, geluid en luchtvervuiling) en de efficiëntie – met name onder druk van de concurrentie – toe, hetgeen terminal operators dwingt na te denken over innovatieve oplossingen, die de ruimte efficiënter gebruiken en tevens tot lagere kosten per containerbehandeling leiden. Een mogelijkheid die zich reeds op enkele plaatsen heeft bewezen in de container terminalindustrie is het automatiseren van processen en het robotiseren van fysieke handelingen. Deze aanpak kan leiden tot een significante reductie van arbeidskosten en een intensivering van het ruimtegebruik met 50%.

Deze beloftes wekken de verwachting dat geautomatiseerde terminals wijd verspreid zouden moeten zijn, hetgeen (nog) niet het geval is. Een reden hiervoor is dat tot nog toe de invoering van dergelijke concepten niet zonder problemen is verlopen. De prestatie van de reeds gerealiseerde terminals is vooralsnog niet conform de gestelde doelen en de terminals hebben te kampen gehad met aanzienlijke opstartproblemen. Veel van deze problemen en de uitblijvende prestaties hangen samen met problemen of beperkingen in de terminal besturingssoftware, zoals case research ons heeft getoond. In detail hebben we de totstandkoming van de ECT Delta-Sealand terminal in Rotterdam geanalyseerd. Deze case leverde de volgende observaties op:

- De frequentie van storingen is onderschat, hetgeen heeft geleid tot onvoldoende procedures in geval van een storing.
- De tijdsdruk in het project heeft ertoe geleid dat er hoofdzakelijk aandacht was om het systeem aan het draaien te krijgen, in plaats van om de prestatiedoelstellingen te bereiken. Als gevolg is een aantal functionaliteiten niet geïmplementeerd zoals beschreven in de specificaties.
De interfaces tussen diverse componenten van het besturingssysteem waren een weerspiegeling van een onderhandelingsproces, meer dan van een logisch ontwerp.

De operatie na ingebruikstelling bleek anders te verlopen dan gepland.

Doordat de terminal niet vanuit een holistische ontwerpgedachte is ontworpen, is op diverse punten het systeem gesuboptimaliseerd.

Daarnaast hebben we de volgende observaties aan andere uitgevoerde cases en literatuur ontleend:

- Veelal bestaat er een kloof tussen het functioneel ontwerp van een terminal en het technisch ontwerpen en de implementatie bij de software realisatie.

- Tijdens het ontwerp van equipment en de besturingssoftware is onvoldoende sprake van een geïntegreerd ontwerp, noch van een interactief ontwerpproces, hetgeen leidt tot suboptimalisatie van componenten. Zelfs binnen het equipmentontwerp is veelal nog sprake van een gefragmenteerd ontwerp.

- Er bestaat onvoldoende aandacht voor de interactie tussen de operator en het te bedienen geautomatiseerde systeem.

- Er bestaat een kloof tussen hoog niveau strategische doelstellingen, zoals doorzetvolumes en scheepsservicetijden, en operationele doelen zoals kraanproductiviteit en truck servicetijden.

- Er bestaan onvoldoende gereedschappen die inzicht verschaffen in de operatie op geautomatiseerde terminals en hun besturingssystemen.

- Er bestaat (nog) geen kant-en-klar besturingsysteem voor geautomatiseerde container terminals, waardoor het projectrisico veel groter is.

- In de beschikbare analysetools is onvoldoende integratie tussen de kostenmodellen en de modellen om prestatie van de terminal te onderzoeken.

- In huidige ontwerp aanpakken is het proces na ingebruikstelling niet opgenomen, bezijden monitoring en ex-post evaluatie.

De situatie waarin container terminals zich momenteel bevinden is derhalve niet eenvoudig. Enerzijds zijn nieuwe oplossingen noodzakelijk om aan de toenemende eisen wat betreft prestatie, kosten en milieu-technische aspecten te voldoen, anderzijds bestaan er nog geen standaard oplossingen voor geautomatiseerde terminals – die wel het perspectief bieden om aan deze verzwaarde eisen te voldoen –
noch bestaan er geëigende manieren om het een gering risico dergelijke terminals te realiseren. Het perspectief van achterblijvende prestaties en omvangrijke opstartproblemen is voor veel terminals onaantrekkelijk, hetgeen vraagt om een ontwerpaanpak die de specifieke problemen van geautomatiseerde terminals adresseert. Dit heeft ons tot de volgende onderzoeksdoelstelling gebracht:

Het ontwerp van een ontwerpaanpak voor gerobotiseerde container zeeterminals, die de specifieke eigenschappen van dergelijke terminals alsmede de specifieke omgevingsfactoren expliciet meeneemt.

De gevolgde onderzoeksmethode is de inductief hypothetische aanpak. Gebaseerd op de inductieve cases hebben we een raamwerk van ontwerprichtlijnen gedefinieerd, op basis waarvan we een in detail uitgewerkte ontwerpaanpak hebben ontwikkeld. Onderdeel van de aanpak is een uitgebreide suite van simulatiemodellen, die in een groot aantal projecten is toegepast, gevalideerd en toegepast (Dobner et al., 1999, 2000, 2001, 2002, 2003, 2004). De structuur van modellensuite is bouwsteen georiënteerd (Verbraeck et al., 2002), hetgeen inhoudt dat elke modelcomponent gedefinieerd is door zijn interface, terwijl de implementatie daarbinnen kan variëren afhankelijk van de toepassing waarvoor het model wordt gebruikt. Gedurende het gehele ontwerpproces blijven de interfaces identiek, terwijl het daarbinnen gemodelleerde gedrag verandert onafhankelijk van andere componenten.

De ontwerpaanpak en de daarvan onderdeel uitmakende modellen zijn toegepast in een aantal toetsende case studies, waarin we ervaringen hebben verzameld met betrekking tot de bruikbaarheid en toegevoegde waarde van de aanpak. Daarnaast hebben we een expert panel ondervraagd met betrekking tot het nut van de belangrijkste ontwerprichtlijnen.

Als basis voor onze ontwerpaanpak hebben we de reeds beproefde simulatie aanpak gekozen (Sol, 1982), die zich heeft bewezen als geschikte aanpak voor complexe problemen.

De aanpak is voorts gebaseerd op vier te onderkennen activiteiten binnen het ontwerpproces:

− Functioneel ontwerp.
− Technisch ontwerp.
− Implementatie.
− Ingebruikstelling en operatie.
Gedurende elke activiteit stellen we voor een simulatie aanpak toe te passen, hetgeen betekent dat voor alle ontwerpvragen een probleemoplossende **cycle** wordt gebruikt. Deze aanpak leunt sterk op het gebruik van modellen om zowel de huidige als de mogelijke alternatieve oplossingen in kaart te brengen en te analyseren. Hiervoor is de reeds genoemde suite van modellen ontwikkeld, die gedurende elke activiteit kan worden ingezet, zelfs tijdens de operatie voor de *fine-tuning* van het systeem, of wanneer veranderende externe factoren heroverweging van de oplossing vereisen.

De volgende ontwerprichtlijnen vormen de basis van de ontwerpaanpak:

- Gebruikmaking van een objectgeoriënteerde bril op de werkelijkheid.
- Toepassing van een holistische, gelaagde modellering van processen op een terminal.
- Identieke structurering van het model en het werkelijke systeem.
- Expliciete modellering van onzekerheid en procesvariatie.
- Gebruikmaking van de reeds bestaande operationele processen als leidraad voor het ontwerp.
- Integratie van handbediende en geautomatiseerde taken.
- Integratie van hardware- en softwareontwerp.
- Definitie van een breed spectrum van meetbare prestatie-indicatoren om de ontwerpalternatieven te beoordelen.
- Het baseren van beslissingen op de uitgevoerde prestatievergelijkingen.
- Continue monitoring nadat het systeem in gebruik is gesteld.

Deze ontwerprichtlijnen zijn uitgewerkt in een gedetailleerde ontwerpaanpak, die bestaat uit een stapsgewijze, iteratieve aanpak om de terminalcomponenten te ontwerpen, een suite van modellen om beslissingen te ondersteunen tijdens deze ontwerpstappen, en een wijze waarop het gehele proces kan worden beheerst.

De stapsgewijze bestaat uit de volgende hoofdstappen:

- Definitie van de functie van de terminal, de beoogde doorzetcapaciteit en de dienstverlening die de terminal moet vervullen.
- Ontwerp van de belangrijkste componenten van de terminal, dat wil zeggen de kadelengte, de terminal afmetingen, de layout van het stack, het transport-, overslag, en opslagsysteem, en het logistieke concept.
- Ontwerp van het equipment en het besturingssysteem.
Na het functioneel ontwerp van de terminal, kunnen alle componenten worden uitgewerkt in het technisch ontwerp en specificatie, waarnaar implementatie volgt. In praktijk is deze gesuggereerde sequentie niet realiseerbaar: altijd zullen diverse iteratieslagen noodzakelijk zijn, waarbij de modellen van ondersteuning zijn.

Na de implementatie kan het systeem worden afgenomen en in gebruik worden gesteld. Gedurende dit proces wordt getest of elke component in interactie met andere componenten voldoet aan de gestelde eisen. Wanneer deze activiteit succesvol is afgerond, kan de operatie van start gaan. Gedurende de eerste periode kan verdere verfijning van functionaliteiten plaatsvinden.

Het gehele proces, hier in een notendop beschreven, volgt een simulatie aanpak, hetgeen betekent dat in elke stap een beoordeling plaatsvindt om de kwaliteit in termen van de gestelde prestatiedoelstellingen vast te stellen. Hierdoor bestaat in alle stadia van het proces een beeld van de mate waarin de doelstellingen worden bereikt.

Gedurende het functionele ontwerp, worden vragen beantwoord zoals de benodigde kadelevenge, het vereiste aantal kadekranen, de gevraagde opslagcapaciteit, en de capaciteit van de afhandeling van diverse modaliteiten zoals rail en vrachtwagens. Daarna wordt binnen deze capaciteitsgrenzen het overslag-, opslag-, en transportsysteem ontworpen, inclusief het logistieke concept dat voor de tijdige en volgordelijke aansturing zorgt.

Tijdens het technisch ontwerp wordt het besturingssysteem ontwikkeld door middel van prototyping van algoritmen, specificatie van parameters en configuratie van de terminal gebruikmakend van de model suite. Aangezien er vooralsnog geen kant-en-klaar systeem bestaat, dient dit voor een groot deel te worden ontworpen en ontwikkeld.

De simulatie aanpak vervult tijdens de implementatie en het testen de rol van testomgeving voor zowel hardware (equipment) als softwarecomponenten. Aangezien stapsgewijs componenten beschikbaar komen als realisatie, en pas aan het eind van het ontwikkeltraject een volledig systeem beschikbaar is, kan tussentijds de modelomgeving de overige componenten afbeelden, zodat componenten in een volledige omgeving kunnen worden getest op prestatie en technische robustheid. De modellen dienen in voldoende mate de dynamiek van een werkelijke operatie te
bevatten en kunnen als zodanig een veeltal van scenario’s afbeelden waaronder componenten kunnen worden beproefd.

Gedurende de ingebruikstelling en tijdens de operatie kunnen dezelfde modellen worden gebruikt om knelpunten te vinden, indien het systeem niet zo functioneert of presteert als gewenst. Daarnaast kan de modelomgeving als maatstaf voor de productieomgeving dienen, hetgeen wel vereist dat in de modelomgeving voldoende detail is meegenomen. In dat geval dient de productiesoftware dezelfde prestatie te leveren als in de modelomgeving wordt bereikt.

Om de bovengenoemde aanpak te realiseren met behulp van modellen dient de modelomgeving eenzelfde structuur te hebben als het werkelijke systeem. Dit maakt het mogelijk componenten in modelvorm en productievorm uit te wisselen zonder de interfaces en/of gedrag aan te passen, hetgeen de kwaliteit ten goede komt.

De ontwerpaanpak is toegepast in een aantal test cases. Ten eerste in het software herontwerp traject van de ECT Deltaterminals, waarin de aanpak is toegepast op de analyse van het huidige systeem, het functioneel en technisch ontwerp, alsmede de ondersteuning van de software implementatie. Aangezien wij zelf deel uitmaakten van het project team kan het onderzoek als action research worden geclassificeerd. In dit project bleek de aanpak goed toepasbaar wat betreft het werken naar de gestelde prestatiedoelstellingen, alsmede in het vinden van problemen in de software. Daarnaast is de feedback van de teams belast met het functioneel en technisch ontwerp, alsmede de software ontwikkelteams zeer positief.

De tweede case waarin de aanpak is toegepast betreft het detailontwerp van een stacking systeem met zeer hoge stack dichtheid: een geautomatiseerde rolbrugkraan. In dit project is de simulatie aanpak ingezet om te komen tot een uitgebalanceerd, geïntegreerd ontwerp van alle kraancomponenten. Alle componenten van de kraan, zowel wat betreft de diverse equipmentonderdelen, als de besturingsregels, zijn in detail gemodelleerd. Dit heeft het mogelijk gemaakt de kraan als geheel te beoordelen, waarbij de kraan ook als onderdeel van een totale terminal is beoordeeld. De aanpak heeft geleid tot een kostenefficiënt en productief ontwerp.

Tot slot is de filosofie achter de aanpak voorgelegd aan een tiental experts op het gebied van simulatie en/of ontwerp van container terminals via een gestandaardiseerde vragenlijst. Uit deze evaluatie blijkt dat de experts de aanpak
overwegend als duidelijke bijdrage zien aan de kwaliteit van het ontwerpproces van geautomatiseerde container terminals.

Het onderzoek heeft geleid tot de volgende conclusies met betrekking tot de onderzoeksvragen:

− Om tot een effectieve ontwerpaanpak te komen dienen de modellen een goede afbeelding te bevatten van het terminalbesturingssysteem, aangezien dit een belangrijke sleutelrol vervult. Om het terminalbesturingssysteem van een gerobotiseerde terminal af te beelden is – in vergelijking met een handbedienende terminal – naast een afbeelding van de hoogniveau besturing, ook een afbeelding nodig van de software die het gerobotiseerde equipment bestuurt, waaronder functionaliteit als routering, en vermijding van botsingen en deadlocks. Naast de weergave van het besturingssysteem op functioneel niveau, is het noodzakelijk ook het gedrag in technische zin mee te nemen, i.e. de duur van planningsprocessen, reactietijden, asynchroniteit van processen, en communicatievertragingen.

− Om het risico van prestatieverlies in de operatie te beperken, richt de aanpak die wij voorstaan zich op het bereiken van de prestatiedoelstellingen gedurende het gehele proces, dat wil zeggen inclusief de implementatie. Daarnaast biedt de aanpak de mogelijkheid software in een vroeg stadium te testen onder realistische omstandigheden, die in voldoende mate de dynamiek van de operatie weergeven. Op deze wijze kunnen de complexe interacties tussen diverse componenten worden getest. Verder biedt een testomgeving die bestaat uit modelelementen en productiecomponenten de mogelijkheid operators te trainen in een virtuele omgeving waarin realistische operaties plaatsvinden.

− Om te zorgen dat het inzicht dat gedurende het gehele ontwerpproces wordt verschaft door de modellen, de validiteit van de modellen is van groot belang. In de situatie dat het gaat om een innovatief systeem, met name op het gebied van het besturingssysteem, kunnen de modelelementen worden beschouwd als prototype van het te bouwen productiesysteem. Tijdens de implementatie dient men zeker te stellen dat de implementatie in het productiesysteem minimaal dezelfde functionaliteit heeft als het prototype, hetzij door middel van een intensieve samenwerking met de leverancier van het softwaresysteem,
hetzij door gedetailleerde specificaties van de software gecombineerd met prestatie-eisen en –tests van de software tijdens oplevering.

De ontwerpaanpak dient onafhankelijk te zijn van de specifieke eigenschappen van het te realiseren systeem om te zorgen dat de vergelijkingen worden gemaakt op basis van dezelfde prestatie-indicatoren. In principe zijn de geformuleerde ontwerprichtlijnen valide voor elk container terminal ontwerpproces. Daarnaast bevat de modelsuite een afbeelding van de meest gebruikelijke systeem die hetzij reeds zijn toegepast of in de pijplijn zitten. Het kan echter zo zijn dat een nieuw systeem de ontwikkeling van nieuwe modelcomponenten vergt. Echter de structuur in de modellen is zo opgezet dat nieuwe systemen met geringe waarschijnlijkheid leiden tot een aanpassingen van de structuur, hetgeen inpassing eenvoudig maakt.

Met betrekking het toepassen van een simulatie aanpak gedurende alle activiteiten van het ontwerp- en realisatieproces, komen we tot de volgende observaties:

- Ten eerste wordt een simulatie aanpak nog steeds hoofdzakelijk geassocieerd met indicatieve resultaten in plaats van precieze beoordelingen, hetgeen zou betekenen dat ze niet geschikt is voor precieze activiteiten als prototyping en software testen. Echter, wij zijn van mening dat indien modellen een voldoende detailniveau bevatten, ze wel degelijk geschikt zijn voor laatstgenoemde activiteiten. In de praktijk wordt echter simulatie niet vaak gebruikt voor dergelijke doeleinden (zie voor vergelijkbare toepassingen Wysk, 2001).

- Ten tweede vergt een simulatie aanpak meer tijd; met name in het begin van het ontwerpproces. Deze extra tijd draagt echter bij aan de kwaliteit van het ontwerp en daarmee de reductie van het risico van de investering. In sommige gevallen is deze extra tijd echter niet beschikbaar. Vooral tijdens het ontwikkelen van de besturingsssoftware is de tijd die nodig is voor het uittesten van alternatieve algoritmen soms te lang, zodat beslissingen toch meer op gevoel dan op systematische analyse worden genomen.

- Ten derde is het onze bevinding dat de professionele omgeving in container terminals hoofdzakelijk is gericht op de operatie. Alhoewel een trend van stijgend opleidingsniveau is waar te nemen worden veel managementposities op container terminals bekleed door mensen die uit de operatie komen. Over het algemeen zijn deze personen minder vertrouwd met een wetenschappelijke,
systematische aanpak bij het nemen van beslissingen omtrent het ontwerp en/of de operatie. Eerder vertrouwen zij op waarnemingen en ervaring. In veel gevallen voert dit tot goede resultaten, maar het risico dat een verkeerde beslissing met de bijbehorende negatieve gevolgen wordt genomen is relatief groot. Bovendien, wanneer het beslissingen betreft met betrekking tot innovatieve systemen, zijn ervaringen uit het verleden meestal geen goede raadgever.

In de twee test cases – alsmede in een aantal ontwerptrajecten waaraan we hebben deelgenomen – hebben we een aantal ervaringen opgedaan met betrekking tot de ontwerprichtlijnen en de aanpak. Hieronder staan deze beknopt weergegeven:

- De aanpak heeft bewezen toepasbaar te zijn in zowel trajecten waar het de ontwikkeling van nieuwe terminals betrof, als uitbreidingen van bestaande faciliteiten, als verbeteringen van bestaande operaties. De inhoud en de overwogen oplossingen hangen af van het type project, maar wat betreft methodologie blijkt de simulatie aanpak in alle fasen van het ontwerp- en realisatieproces toepasbaar. Zelfs in projecten waar een besturingssysteem wordt gerealiseerd dat sterk leunt op productiesoftware, blijken veel vragen door middel van een simulatie aanpak te kunnen worden beantwoord.

- Een cruciale meerwaarde van de simulatie aanpak zijn de doorkijkjes naar het technisch ontwerp en de implementatie die door middel van de modellen kunnen worden gegeven. Het in detail uitwerken (in een modelomgeving) van specifieke componenten – equipmentspecificaties, besturingsalgoritmen – geeft feedback aan het functioneel ontwerpproces en creëert een beter beeld van het eindproduct. Met name wanneer het besturingssysteem wordt gebouwd aan de hand van de als prototype ontwikkelde oplossingen in de modelomgeving, bieden de modellen in een vroeg stadium reeds een duidelijk beeld waar het project zal eindigen.

- Door de overeenkomsten tussen verschillende terminals, is bovengenoemde prototyping functionaliteit van grote meerwaarde doordat in korte tijd een beeld kan worden geschetst van het nieuwe systeem. Het toepassen een bouwsteen georiënteerde aanpak is hier van grote meerwaarde om componenten te kunnen hergebruiken, zonder de structuur drastisch aan te passen.
De initiële keuze om de modelsuite bouwsteen georiënteerd op te zetten, heeft in grote mate bijgedragen aan de snelheid waarmee modellen kunnen worden ingezet en daarmee aan de waarde van een simulatie aanpak. Het maakt het mogelijk ook in een fase waarin snel tot antwoorden moet worden gekomen een simulatie aanpak geschikt is. Daarnaast maakt deze aanpak het mogelijk de structuur van de modellen gedurende het gehele proces constant te houden; in de beginfase is de implementatie binnen de gedefinieerde interfaces op een hoger abstractieniveau, in een latere fase wordt deze implementatie vervangen door een meer gedetailleerde afbeelding. In sommige toepassingen kan het zelfs zijn dat sommige componenten veel gedetailleerder zijn dan andere, hoewel ze in een modelomgeving met elkaar interacteren. Deze aanpak verkleint de ontwikkeltijd en biedt de mogelijkheid met hogere snelheid te experimenteren.

Het gebruik van een simulatie aanpak stimuleert de focus op het behalen van de prestatiedoelstellingen. Aangezien het merendeel van complexe projecten zoals het realiseren van (geautomatiseerde) container terminals onder tijdsdruk staat of komt te staan, ligt het in de lijn der verwachting dat de focus verschuift van prestatiedoelstellingen naar het werkend krijgen van het systeem, zoals ook in het ECT project eind jaren 80 is gebeurd. Bewust wordt dan de aandacht verlegd naar het operationeel krijgen in plaats van het bereiken van de prestatiedoelstellingen. Dit draagt het gevaar met zich mee dat in latere fases deze doelstellingen niet meer bereikbaar zijn door bepaalde versimpelingen. De mogelijkheid deze afweging te maken met behulp van een modelomgeving, maakt dat dergelijke beslissingen in ieder geval bewust worden genomen, de consequenties kennisne. Het belang van een valide modelomgeving dient hierbij nogmaals te worden onderstreept.

In aanvulling op het vorige punt biedt de simulatie aanpak niet alleen de mogelijkheid het systeem te beoordelen op prestatie, maar tevens om het systeem onder een veeltal van operationele omstandigheden te testen op zijn functioneren. Dit betekent dat de kans op grote negatieve consequenties van allerlei onvoorziene samenlopen van omstandigheden, zoals storingen, in grote mate wordt gereduceerd, waardoor het risico van het project wordt verkleind. Tevens kan het delen van deze resultaten met de toekomstige operators van de
terminal, leiden tot veel geringere opstartproblemen, omdat de bekendheid met de problematiek reeds aanwezig is.

Een voorwaarde voor succes van een simulatie aanpak is de samenwerking en coördinatie tussen het team dat zich met de simulatie aanpak bezighoudt en de teams die zich met de implementatie bezighouden. Het is onze ervaring dat dit niet altijd eenvoudig is om de navolgende redenen. Ten eerste hebben beide teams verschillende doelstellingen. De prototypes in de modelomgeving zijn ontwikkeld met het oog op het behalen van de prestatiedoelstellingen, en niet op het realiseren van een stuk productiesoftware, waarop het implementatieteam zich richt. Hoewel een en ander verenigbaar zijn, is het niet altijd gemakkelijk het prototype om te zetten naar productiesoftware. De tweede reden is de parallelliteit tussen de software ontwikkeling en de simulatie aanpak ter ontwerp en verbetering van het software ontwerp. Om deze ontwikkelingen synchroon en uitwisselbaar te houden is het noodzakelijk regelmatig ideeën en concepten uit te wisselen, anders dreigt het gevaar dat de trajecten uit elkaar lopen.

Tot slot een blik in de toekomst: welke vragen in de lijn van dit onderzoek liggen op ons te wachten? Ten eerste zien we als uitbreiding op de monitoring functie die een simulatie aanpak kan vervullen tijdens de operatie, een mogelijkheid dezelfde modellen in te zetten om vooruit te kijken en hiermee real-time besluitvorming te ondersteunen. Hiermee kan beter worden gereageerd op onverwachte gebeurtenissen en kunnen de diverse mogelijkheden worden geëvalueerd op basis van de actuele situatie. De uitkomst kan dan weer als input dienen voor het productiesysteem. Hoewel dit een typisch voorbeeld is van een simulatie aanpak, stelt het zwaardere eisen aan de modelomgeving, door de noodzaak data vanuit de productieomgeving te laden, alsmede in acceptabele tijd tot een resultaat te komen. Op dit moment zijn de modellen die deel uitmaken van onze modelsuite hiervoor te traag: dergelijke vraagstellingen kosten snel een dag experimenteren in plaats van enkele minuten. Echter wanneer de toekomstige hardware snellere uitkomsten mogelijk maakt, zien wij hierin een waardevolle toevoeging van de simulatie aanpak. Een andere mogelijkheid is om functionaliteit te distribueren over diverse computers, hetgeen wederom bevestigt dat een bouwsteen georiënteerde aanpak de juiste is, aangezien hierdoor distributie van functionaliteit mogelijk wordt gemaakt.
Een tweede toepassing van een simulatie aanpak die wij in de nabije toekomst zien en van toegevoegde waarde achten is het zogenaamde *reverse simulation*. Het naspelen van operaties in het verleden met veranderde instellingen om te kijken op welke manier de operatie beter zou zijn verlopen. Deze toepassing vereist een gedetailleerde registratie van alle gebeurtenissen die relevant zijn voor de prestatie, zoals storingen, en veranderende informatie. Op basis van het naspelen van operaties kan tot een betere instelling van besturingsparameters worden gekomen voor specifieke operaties en/of omstandigheden. Tevens kan veel beter worden verklaard waarom een bepaalde prestatie is gerealiseerd, hetgeen weer kan bijdragen aan de verbetering van het systeem als geheel. De kracht van de vergelijking is dat exact dezelfde operatie wordt nagespeeld, zodat de uitkomsten zeer representatief zijn voor toekomstige vergelijkbare operaties.
10.6 Curriculum Vitae Yvo A. Saanen

Yvo Saanen, born March 1973 at The Hague, followed high school at the Dalton Scholengemeenschap at The Hague. In 1991 he continued his education in Delft, where he graduated (MSc) at the faculty of Systems Engineering, Policy Analysis and Management at Delft University of Technology in 1996 on a master thesis about decision-making in large infrastructural projects. Immediately after graduation, he started his own company together with two companions. Since then, the company grew to 16 people, and has customers across Europe and the United States. Yvo Saanen is one of the two managing directors of the company. In parallel, he performed his PhD research at Delft University of Technology, commencing 1999. Since 2003, Yvo Saanen is also part-time assistant professor at Delft University of Technology, teaching simulation and logistics. Yvo Saanen is married and has one son.