

LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
FACULTY OF TECHNOLOGY MANAGEMENT
DEGREE PROGRAMME IN INFORMATION TECHNOLOGY

Antti Hovi

**Technology roadmap for new sales support and management
system for crane manufacturing company**

Examiners: Professor Heikki Kälviäinen
M.Sc. Tero Roivainen

Instructor: M.Sc. Tero Roivainen

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Degree Programme in Information Technology

Antti Hovi

Technology roadmap for new sales support and management system for crane manufacturing company

Master's thesis

2008

133 pages, 14 figures, 1 table and 8 appendices

Examiners: Professor Heikki Kälviäinen
M.Sc. Tero Roivainen

Keywords: Sales system, Systems design, Sales configurator, Crane industry

Sales configurators are essential tools for companies that offer complicated case specifically crafted products for customers. Most sophisticated of them are able to design an entire end product on the fly according to given constraints, calculate price for the offer and move the order into production. This thesis covers a sales configurator acquisition project in a large industrial company that offers cranes for its customers. The study spans the preliminary stages of a large-scale software purchase project starting from the specification of problem domain and ending up presenting the most viable software solution that fulfils the requirements for the new system.

The project consists of mapping usage environment, use cases, and collecting requirements that are expected from the new system. The collected requirements involve fitting the new sales system into enterprise application infrastructure, mitigating the risks involved in the project and specifying new features to the application whilst preserving all of the admired features of the old sales system currently used in the company.

The collected requirements were presented to a number of different sales software vendors who were asked to provide solution suggestions that would fulfil all the demands. All of the received solution proposals were exposed to an evaluation to determine the most feasible solutions, and the construction of evaluation criteria itself was a part of the study. The final outcome of this study is a short-list of the most feasible sales configurator solutions together with a description of how software purchase process in large enterprises work, and which aspects should be paid attention in large projects of similar kind.

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tietotekniikan koulutusohjelma

Antti Hovi

Teknologiakartointu myynnin tuki ja -hallintajärjestelmälle nostolaitteita valmistavassa teollisuusyrityksessä

Diplomityö

2008

133 sivua, 14 kuvaa, 1 taulukko ja 8 liitettä

Tarkastajat: Professori Heikki Kälviäinen
DI Tero Roivainen

Hakusanat: Myyntijärjestelmä, Järjestelmäsuunnittelu, Myyntikonfiguraattori, Nosturiteollisuus

Keywords: Sales system, Systems design, Sales configurator, Crane industry

Myyntikonfiguraattorit ovat tärkeitä työkaluja yrityksissä, jotka tarjoavat vaativia asiakaskohtaisesti räätälöityjä tuotteita loppuasiakkailleen. Hienostuneimmat näistä pystyvät ajonaikaisesti suunnittelemaan annettujen rajoitusten mukaisia lopullisia tuotteita, laskemaan niille hinnat ja siirtämään syntyneet tilaukset tuotantoon. Tässä työssä käydään läpi myyntikonfiguraattorin hankintaprojekti suuressa nostolaitteita valmistavassa teollisuusyrityksessä. Työ kattaa suuren tietojärjestelmähankintaprojektin alkaen ongelma-alueen kartoittamisesta ja päätyen lopulta sopivimman, vaatimukset täyttävän, järjestelmän valintaan.

Projekti koostuu käyttöympäristön ja käyttötapauksen kartoittamisesta, sekä uudelta järjestelmältä edellytettävistä ominaisuuksista. Kerätyt vaatimukset käsittävät myyntijärjestelmän integroimisen yrityksen muihin järjestelmiin, riskien lieventämisen ja uusien ominaisuuksien lisäämisen järjestelmään siten, että yrityksen käytössä olevan vanhan myyntijärjestelmän parhaat ominaisuudet säilyvät myös tulevaisuudessa.

Vaatimukset esitettiin myyntijärjestelmiä tarjoaville ohjelmistoyrityksille, joilta pyydettiin ehdotelmia edellytykset täyttävästä järjestelmästä. Saadut ehdotelmat asetettiin vertailuun, jossa pyrittiin löytämään toteuttamiskelpoisimmat vaihtoehdot seuraavaksi myyntijärjestelmäksi, vertailukriteereiden valinnan ollessa myös yksi osa tätä tutkimusta. Työn lopputuloksena syntyi lopullinen valintalista sopivimmista järjestelmätoimittajista sekä kuvaus tietojärjestelmän hankintaprosessin toiminnasta suuressa yrityksessä ja ulottuvuuksista, jotka tulisi huomioida vastaavissa projekteissa.

PREFACE

When I first learned about my topic little did I know that the technology evaluation, which I initially thought would be a consultation project about programming languages and interfaces, eventually took me to a journey through knowledge systems and their industrial applications. Sales configurators themselves turned out to be not just any business applications; instead they were proven to be complex artificial intelligence systems with serious academic research behind them.

I would like to express my appreciation to all the people that contributed this project and made it possible: Tero Roivainen, manager of sales systems at Konecranes who lifted me up to perceive wider aspects of the project, Heikki Kälviäinen, my professor and instructor during this thesis who did not let me to forget about the academic background of the study.

The greatest gratitude, however, I would like to point to the different sales configurator vendors who took the trouble of travelling around the world presenting the configurator solutions that contributed to this thesis. I could not possibly thank you enough for all your time and efforts.

Hämeenlinna, November 3rd, 2008

Antti Hovi

“Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.”

-Antoine de Saint-Exupéry

CONTENTS

1 INTRODUCTION	4
1.1 Background.....	4
1.2 Objectives and restrictions of this thesis	5
1.3 Structure of this thesis	6
2 COMPLEXITY OF CRANE MANUFACTURING	8
2.1 Konecranes Corporation in general	8
2.2 Standard lifting business area	9
2.3 Multibranding	10
2.4 Mass customization and configuration	12
2.5 Engineering-to-order	13
2.6 Enterprise application integration.....	14
2.6.1 Enterprise resource management system	15
2.6.2 Product data management system	16
2.7 Corporate IT strategy.....	17
2.7.1 Future of design platform software	18
2.7.2 Service oriented architecture meets IBM WebSphere Portal.....	19
3 CURRENT SALES SYSTEMS	22
3.1 Sales configurators in literature.....	22
3.1.1 General approach towards configuration problem.....	22
3.1.2 Implementation of configurator systems.....	26
3.1.3 Implementation of knowledge base	28
3.2 Current sales configurator	33
3.2.1 Architecture of current sales configurator.....	35
3.2.2 User groups and use cases of current sales configurator.....	38
3.3 Other sales systems.....	40
3.4 Sales processes	42
3.4.1 Sales processes SP11 and SP12	42
3.4.2 Sales process SP13	43
4 REQUIREMENTS ANALYSIS	45
4.1 Business study conducted for requirements gathering	45
4.2 Identified risk factors.....	48
4.3 Selected key requirements	53
4.4 IEEE STD 830 Software Requirements Specifications.....	55
4.5 Brief supplementary SRS outline for sales system project.....	57
4.5.1 Description of desired functionality.....	57
4.5.2 External interfaces and information exchange.....	59
4.5.3 Constraints and limitations.....	61
5 SUGGESTION FOR SYSTEM STRUCTURE	63

5.1 Basic structure	63
5.1.1 Presentation tier in detail.....	65
5.1.2 Logic tier in detail	68
5.1.3 Database tier in detail.....	69
5.2 Meeting key requirements	70
6 ALTERNATIVE SOLUTIONS FOR NEW SALES SYSTEM	75
6.1 Sales configurator markets	75
6.2 Methods for finding the alternatives.....	79
6.3 Solution suggestion by Wapice	81
6.4 Solution suggestion by NSD Consulting.....	84
6.5 Solution suggestion by Tacton Systems	87
6.6 Solution suggestion by Oracle Siebel.....	90
6.7 Solution suggestion by Perspectix.....	91
6.8 Solution suggestion by Cincom.....	94
6.9 Solution suggestion by ACBIS / Epos.....	96
6.10 Other solution suggestions.....	98
7 EVALUATING SOLUTIONS.....	100
7.1 Finding the best solution.....	100
7.2 Evaluation criteria.....	102
7.3 Evaluating solutions	105
7.4 Initial impressions of different architectures	106
7.5 Result of evaluation	109
8 CONCLUSIONS.....	110
8.1 Further actions	110
8.2 Final thoughts	112
REFERENCES	114
APPENDICES	

ABBREVIATIONS

ABAP	Advanced Business Application Programming Language
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BOM	Bill Of Materials
CAD	Computer Aided Design
COM	Component Object Model
CRM	Customer Resource Management
DLL	Dynamic-link Library
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
ERM	Enterprise Resource Management
GCM	Global Company Master
GPF	Generic Product Family
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JNI	Java Native Interface
MDM	Master Data Management
MFC	Microsoft Foundation Classes
MSI	Microsoft Installer
OCL	Object Constraint Language
ODBC	Open Database Connectivity
PDM	Product Data Management
PLM	Product Lifecycle Management
RAD	Rapid Application Development
SaaS	Software As a Service
SOA	Service Oriented Architecture
SQL	Structured Query Language
SRS	Software Requirements Specification
SSL	Secure Socket Layer
UML	Unified Modelling Language
W3C	World Wide Web Consortium
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

1 INTRODUCTION

1.1 Background

Fast revolving and globalizing world has created ever more increasing pressure for companies to respond to very dissimilar customer demands at an ever-faster pace in the most cost efficient way. The demands have created a concept of mass customizable or configurable products that allow creation of several variants of a single product to meet the specific needs of a customer.

However, not all the products are such by nature that they can be configured easily. A crane is a perfect example of a product that cannot be configured and offered to a customer just by combining different components from pricing catalogue into a single product. Problems arise from the heavy optimization and engineering routines that are needed to calculate several different variables such as strengths of steel structures and modelling the layout of the crane, let alone that most of these variables have a tendency of being interdependent of each other. To make the configuration problem even more complicated, the variables do not all just depend on the internal product options but also on external conditions such as operating environment or even climate conditions.

Konecranes Corporation has addressed the complex configuration problem by building a sales configuration system named Markman 2000. The current sales system has been in effective use for more than a decade and it has managed to satisfy all of the demands it was originally designed for. However, twenty years is a long time in the field of information technology and during this time a set of new requirements have emerged.

Sales configuration system in the context of this thesis will refer to a computer software product that is used by the salesmen to specify customer demands and construct a conceptual model of the product that is offered to customers. Markman, the current configurator, is also able to optimise the offer for the most-efficient construction and send it to production. Therefore it is reasonable to label the current system not only as a sales configurator but instead an entire sales support and management system.

Having served several years laudably, Markman is now facing an inevitable situation in which it needs to be redesigned and renewed. As Markman plays a very significant role in the entire corporation, the renovation will be done using very careful planning. This thesis is a part of the preliminary study that is taken before the actual realisation of

redesign project and focuses on outlining and evaluating the different technological approaches for the new system. [1]

1.2 Objectives and restrictions of this thesis

The objective of this thesis is to find the most feasible new technology platform for the future sales configurator system of standard lifting business area of Konecranes Corporation. This thesis covers and maps existing commercially available solutions and evaluates their suitability for the company. This suitability is deeply influenced by the current business practices and real-life requirements of the crane industry, such as a need of heavy computation in strength calculations.

The term technology platform in this work refers to databases, programming languages, interfaces and integration to other systems of the company, such as Enterprise Resource Management (ERM) and Product Data Management (PDM) systems, as well as different platform level technologies such as several Computer Aided Design (CAD) systems.

Since a perfect commercially available solution is unlikely to exist, and the vendors of such products are few and far between, this technology roadmap will try to answer the problem by splitting the massive sales configurator system into smaller subcomponents and subsystems that are more commonly available. One of the paths that need to be explored is whether the own system development team of the company can solely or partly build some of the subcomponents of new system, and to what extent that can be done.

Other aspects that are being considered are the possibilities to use the system in the offline mode as well as in the online mode. In case that even a partial commercial solution cannot be found, the aim of this work is to suggest a technology platform on top of which the system can be built by the own software development team of the company.

The actual systems design and project planning is, however, left outside of the scope of this work together with in-depth requirements specification, since they are separate projects. Business requirements specification for the sales system is a separate on-going project at the company and it is tightly coupled to the technology evaluation and will thus be briefly covered in chapter four. [1]

1.3 Structure of this thesis

This work is structured in a way that the background and motivation for the entire configurator system are presented from the company's point of view in Chapter 2. The chapter introduces the reader to the company and helps to understand the need and importance of sales configurator. The first three chapters are aimed to assist in understanding the problem domain and explain the importance and motivation behind sales configurator systems. The first half of this thesis is used in determining the qualities that are required from the future sales configurator application, and the second half contains a comparison of different alternatives for the next generation sales system.

Chapter 3 depicts the current sales configuration system that the company is using. For the new sales system design project to succeed, it must be at least to be able to meet all the functionality and features that the current system offers. In addition to that, the chapter also describes the sales processes, the human interaction, and the use cases of the current system as a background for the new system.

Chapter 4 collects all the different requirements for the new system, and forms an outline for requirements analysis by utilising IEEE (Institute of Electrical and Electronics Engineers) Standard 830 to strip apart the findings. The chapter also covers the results of a separate business need analysis that took place prior this study, and all the different requirements are collected into this chapter and presented to various sales configurator vendors as a preliminary specification in the offer stage.

Chapter 5 continues requirements specification outline by introducing an example of a system architecture depicting how the system might be constructed. The purpose of this chapter is to provide a case example of a configurator architecture that would help to clarify the different requirements and to ensure that they get correctly understood. Transferring the software requirements for third party developers is always a task filled with pitfalls and introducing an architecture that would seemingly fulfil all of the requirements is hoped to mitigate the risks of misunderstanding.

Chapter 6 presents the different sales configurator vendors and the software solutions that they are offering. The chapter explains methods that were used in finding the different configurator vendors and presents nine different solution suggestions that appeared to be most feasible ones for the given purposes in greater detail. The sales configurator solutions that are presented here are evaluated in further chapters, and compared against the requirements that were collected earlier.

Chapter 7 introduces different metrics that are used to evaluate the different configurator solutions and depicts the methods how the most feasible ones are picked. Once the final evaluation criteria are selected and assigned with weight factors, the different alternatives are compared against the metrics. Finally the solutions that appear to be most satisfying for the purposes of the company will be presented.

Chapter 8 draws the final conclusions, summarises the project and presents some suggestions for future actions to take place after this study. The chapter concludes the thesis with a review of earlier stages of the project, and offers insights for other projects of similar kind.

2 COMPLEXITY OF CRANE MANUFACTURING

Konecranes Corporation is the leading crane manufacturer in the world, operating in several different fields of crane industry, manufacturing being just one of the operating areas. This chapter provides an in-depth view to the functions and business areas of the corporation that influence the requirements that are presented for the sales system.

2.1 Konecranes Corporation in general

Konecranes Corporation is a company that operates in several areas in hoisting and lifting industry. The corporation is divided into several business area units that handle heavy cranes, crane maintenance services and production of standard cranes –the ones that are offered by the means of productization– distinctively.

The different business areas of the corporation are depicted in Figure 1 which shows three different sections that represent business areas by their sales percentage. The service area focuses on the maintenance and modernization of existing cranes, even the ones that are built by the other crane manufacturers. Heavy lifting area deals with big custom-built cranes, such as harbour or gantry cranes. Standard lifting area handles all of those cranes that can be offered to a customer by the means of mass customization.

In many parts, the sales of the corporation does not only consist of cranes that Konecranes Corporation delivers to its customers, instead it consists also of selling crane components to other differently branded companies, subcontractors and licensors that sell the finished product to end customers. Currently the corporation employs directly more than 8000 people and operates in more than 40 countries. [2]

Growth rate of the company has been rapid in recent years, which has also posed challenges for the entire business infrastructure that also includes the software development area. The company has not only grown organically but also by acquisitions of several smaller crane manufacturing companies leading to a situation where there are several different international manufacturing sites that operate in slightly different manners. Currently the corporation is executing a data harmonization project that aims to integrate the different information systems into a single, more easily controllable framework. The requirements brought up by the harmonization project will also pose challenges for the future development of sales configurator system, which should be taken into account in the requirements analysis. [2, 3, 4]

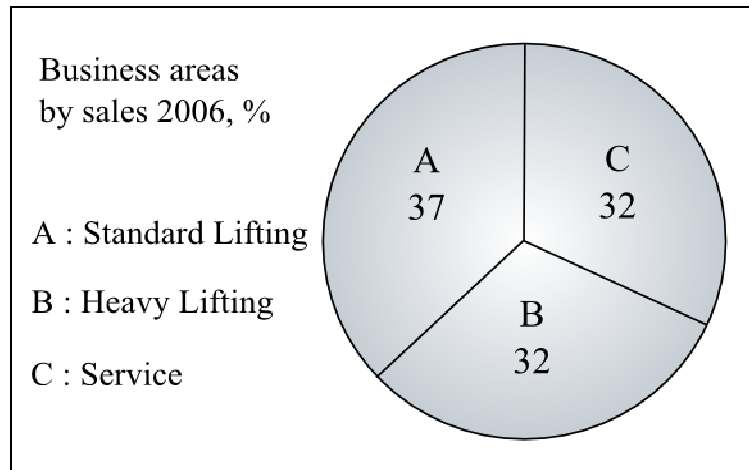


Figure 1. Business areas of Konecranes Corporation [2]

2.2 Standard lifting business area

The standard lifting business area offers cranes that are manufactured by the means of mass customization. Such cranes include, for example, industrial cranes, workstation cranes, chain hoists and wire rope hoists. What separates the standard lifting business area from the heavy lifting area is that it generally focuses on smaller cranes that are used in industrial plants to lift and move cargo within one building or a factory hall. [2]

Industrial cranes are the kind of cranes one may encounter at manufacturing sites such as paper factories or steel factories. Although all the cranes are built by the custom demands reflecting their usage, they still are built mainly from standardized components and thus resemble the general product structure of the business area. Because of the flexible design paradigm, the applications of industrial cranes vary from small semi-automatic assembly lines into the large-scale explosion proof crane installations found at the oilrigs. Konecranes is currently a global technology leader in the field of industrial cranes and owes the status greatly to a modular product structure that it uses. This kind of modular design paradigm is referred to as mass customization among the industry, and it will be presented in greater detail later in this chapter.

Workstation cranes and chain hoists on the other hand are smaller cranes than their industrial counterparts, and they are designed to move small weights of cargo in a limited space. The chain hoists provide, for example, an inexpensive solution for those situations where it is impractical to operate large full-scale crane in order to move items inside a manufacturing cell. The design methods behind the workstation and chain hoists are, however, similar to full scale industrial cranes so that they can be configured and tailored to meet the specific needs of the customer. [5]

Wire rope hoists and cranes components represent the part of business that takes care of selling individual crane components to the customers. Wire rope hoist is the part of the crane that is actually used to lift the loads. However, sometimes it is not necessary or even meaningful for a customer to purchase an entire crane, instead a single hoisting unit may be enough to satisfy the need. A big part of the sales at standard lifting consists of component sales, which is due to the fact that a big part of crane sales in general takes place through licensors and retailers. Also, the company itself consists of several different regional brands that purchase the crane components from other factories. This aspect plays also an important part in the design of the sales configurator and will be explained in following chapters. [5]

Modernization services is the part of the standard lifting business area that offers services to renovate old cranes to meet the modern day standards. This operation does not pose new requirements for the sales configurator. [5]

2.3 Multibranding

Crane industry is very fragmented, and thus the corporation has not only grown organically, but also by buying smaller enterprises that have seemed to bring more value for Konecranes. Whenever a new company is acquired to a part of Konecranes Corporation, it usually comes with an existing brand that generally is worth preserving. All the different brands sell mostly the same components as the mother company, although some of the products are labelled differently and they may be sold using different business practices.

Currently the brands of Konecranes consists of Morris Material Handling in the UK, Verlinde in France, SWF Krantechnik and Stahl in Germany, Meiden Hoists System in Japan, R&M Material Handling and P&H in the United States. The brand Konecranes is mainly, though not exclusively, used in the Scandinavia, Europe, Asia, USA, Canada and Oceania.

The reasons behind having different brands that sell the same cranes may not be obvious at the first sight. However, when examining the entire set of different brands from customer's point of view, it is easier to see the benefits. Verlinde, for example, is one of the oldest crane manufacturers in the world, having served customers for more than 150 years. Verlinde has a strong reputation in France and some of the customers are more willing to buy a new crane from a manufacturer that they already know. Similar benefits of the brand recognition can be applied to other brands as well.

The brands are separated into two entities: alpha brands and beta brands. The alpha brands do not conduct business directly with the end customers; instead they sell crane components to distributors and independent retailers. The beta brands, on the other hand, act more conventional manner and sell differently labelled cranes directly to the end customer. Beta brands do sell components to some extent but, nevertheless, it is not a dominant form of business. Some of the brands, such as Morris UK and Stahl, currently mix the business practices of selling the components and cranes, but in general alpha brands refer to the crane component sales whereas beta brands refer to the selling of entire cranes.

Selling the components even to direct competitors brings an advantage that may not appear obvious at the first sight. The business idea behind selling crane components to the licensors or independent component customers that bid against the company in competition situation is illustrated in Figure 2. When a customer puts a lifting solution out to tender, in the end a big part of cranes that are offered to the customer are built from the parts that are manufactured by Konecranes. Bigger offer coverage increases also the chances of getting a sale, and even if a bid is lost to a licensor or an independent crane manufacturer, the component selling alpha brand may still get the sale of the components. [1, 2]

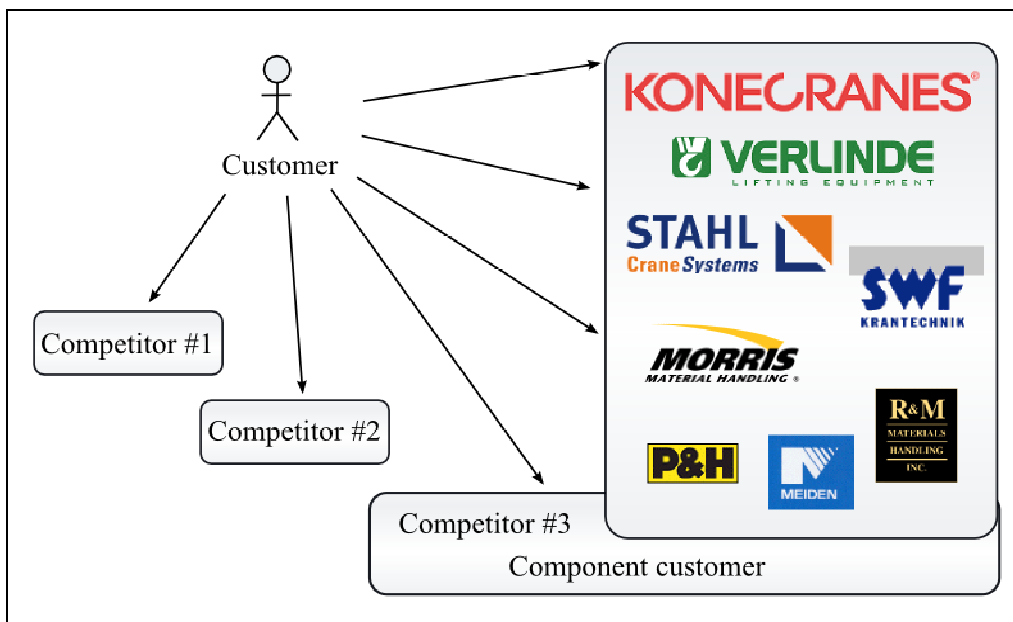


Figure 2. Customer choosing a crane vendor

The different brands pose challenges for the new sales system. All of the brands, licensors and component customers use the same sales system for making their purchase of desired parts. Because of the user base of the system exceeds the boundaries of just one company, the needs of the different brands and brand neutrality ought to be taken

into account. More than 60% of sales in the Standard Lifting Area consist of component sales, therefore it is important to be able to serve all the different types of users equally. [1]

2.4 Mass customization and configuration

What separates the standard lifting area from the other business areas, most notably heavy lifting business area, is the fact that all the cranes offered to the customers are built from the components that are designed in way that they can be combined with a very little engineering effort. As stated earlier, this method of design is called mass customization.

The birth of mass customization took place in the early nineties, when increasing competition between companies challenged them to produce more customer oriented products and services at smaller costs. The companies had an urge to serve their customers by offering them products that suited the specific needs of a small particular customer field without sacrificing any of the profitability. [6]

Term “mass customization” is most often credited to Joseph M. Pine whose similarly named book was published in the early nineties. The fundamental idea behind the term, in addition to ideas presented earlier, is to transform the product development into smaller sub entities that enables faster development pace and overall efficiency thanks to the increased flexibility. Flexible product platform is also able to respond varying market conditions faster, since a single production chain is able to manufacture products for a number of different customer segments. [6]

Mass customization is strongly tied with Just In Time (JIT) inventory strategy that aims to reduce the in-process inventory and the costs that are involved in stocking goods. The JIT strategy works basically in a way that the products are built on-demand instead of manufacturing them in large quantities in advance. With mass customized product this is almost a necessity because parts, called product modules, that are used to build product variants are often customer and order specific. [6]

Configuration is the other side of the mass customization. Before a mass customized product can be offered, the product range must be built and designed in a way that makes it possible to tailor the product to meet the specific needs. When a product, a crane in this case, is designed in way that new product variants can be built from interchangeable modules, the product is called configurable. A configurable product is much more than just a single product; it represents an entire product family that can be

configured to represent a particular product variant. So, when a customer wishes to purchase an industrial crane with the given hoisting loads, transportation dimensions and hoisting heights, a generic product model gets configured in a way that it satisfies the given requirements. This is also what the sales configurator software essentially does. [6, 7]

The modularity in the product design has made it possible to implement new technical innovations into the existing crane models thus eliminating the need of entirely new product design whenever a single new feature is needed. This modular and configurable product range has given the standard lifting business area a clear head position in the market emphasizing the importance of the sales software that is now being renovated.

2.5 Engineering-to-order

The term engineering-to-order may refer to a product that is engineered in advance for the needs of a specific customer, or a product that is engineered once after the order has been made. Not all the possible crane variations can be designed in a way that the complete manufacturing drawings would exist in advance. Some of the cranes that are sold require special engineering efforts and cannot be manufactured simply by putting existing product modules together, for the reason that certain special option modules still need case-specific manual engineering, which is also known as platform level product variation. [8]

The engineering-to-order approach is what separates the crane configuration, for example, from the automotive industry. Other lines of industries rely heavily on the pre-designed components where the number of different product configurations is finite. For a sales configurator, the engineering-to-order method poses an enormous challenge because the configurator has to act as a simple and easy-to-use sales tool, but also as an engineering tool that is able to construct feasible crane variants on the fly. In the previous automotive industry analogy this would equal to a situation where the user would have a possibility to enter a preferred cubic size of the engine into a free text field, and after that the sales configurator would calculate whether the engine fits into the engine bay, compute the positions of engine holding braces and ultimately optimise the entire structure in the most economical way. [8]

The output of the sales configurator is a modular Bill Of Materials (BOM) that defines the structure of a created configuration. A BOM is a list that consists of all the different parts and modules that are required to manufacture a product variant, so that when a customer orders a configured product, the order information is stripped apart and

transformed into internal product codes and part numbers, which can be later used at the production stage in the other enterprise systems. [9, 10, 11]

The term BOM is used in several contexts throughout the industry and Konecranes makes no difference. A full engineering BOM (EBOM) depicts an entire product structure to the smallest nut and bolt. This information is used in the manufacturing, when all the possible details are needed to build the end-product and its components. The EBOM information is stored in a Product Data Management (PDM) system together with several other design documents. When a sales configurator contains the sales oriented information of the product structure, it actually contains a simplified version of the EBOM, which is usually called modular BOM or sales BOM. [7, 12]

One of the challenges when designing a new sales configurator is to explore whether it is possible to re-use the existing EBOM information to create a sales BOM structure. For the time being, EBOM information is not used in construction of sales BOM structure in any automatic way, instead the two separate models are built independently only to reflect one another. Nonetheless, the EBOM cannot be used directly in the sales configurator because the information it contains is engineering and technology oriented and way too specific in detail. However, it might be possible to build a tool that would extract the sales BOM information from the EBOM automatically in the future. [1, 7]

2.6 Enterprise application integration

Enterprise Application Integration (EAI) is a term that is used to describe how the different IT-systems relate to each other within one company. Usually enterprise applications include systems for managing supply chains, customer relationships and product data knowledge. EAI depicts the architecture of how the different large systems communicate with each other and how the information travels through the systems.

Figure 3 shows how the order information moves through different IT-systems throughout the standard lifting business area before the product is shipped. The order initially enters the system from the sales configurator, Markman 2000. From then on, the order information is transferred into the Enterprise Resource Management (ERM) system that forwards the information to the PDM system. The sales configurator also synchronizes the order information to Proactive Crane Offer Manager (ProCom) and Project Follow-up (ProFlow) -systems that are used to monitor the status of the order. The entire sales system block will be covered in more specific detail in chapter three.

When the order enters the PDM system, it still uses the sales BOM-structure and it will be reconfigured to use the EBOM structure. If there is any platform specific product variation involved, it takes place inside the PDM system. Once the product variation is complete, and the PDM system has finished converting the order structure to reflect the EBOM structure, the order will be transferred to the ERM system that sends the order row information to the production. The order information at this point has gotten converted into multiple order rows that represent the individual entities of a crane structure. The PDM system also triggers document-folding procedure that will automatically generate the user's manuals and other delivery documents for a crane.

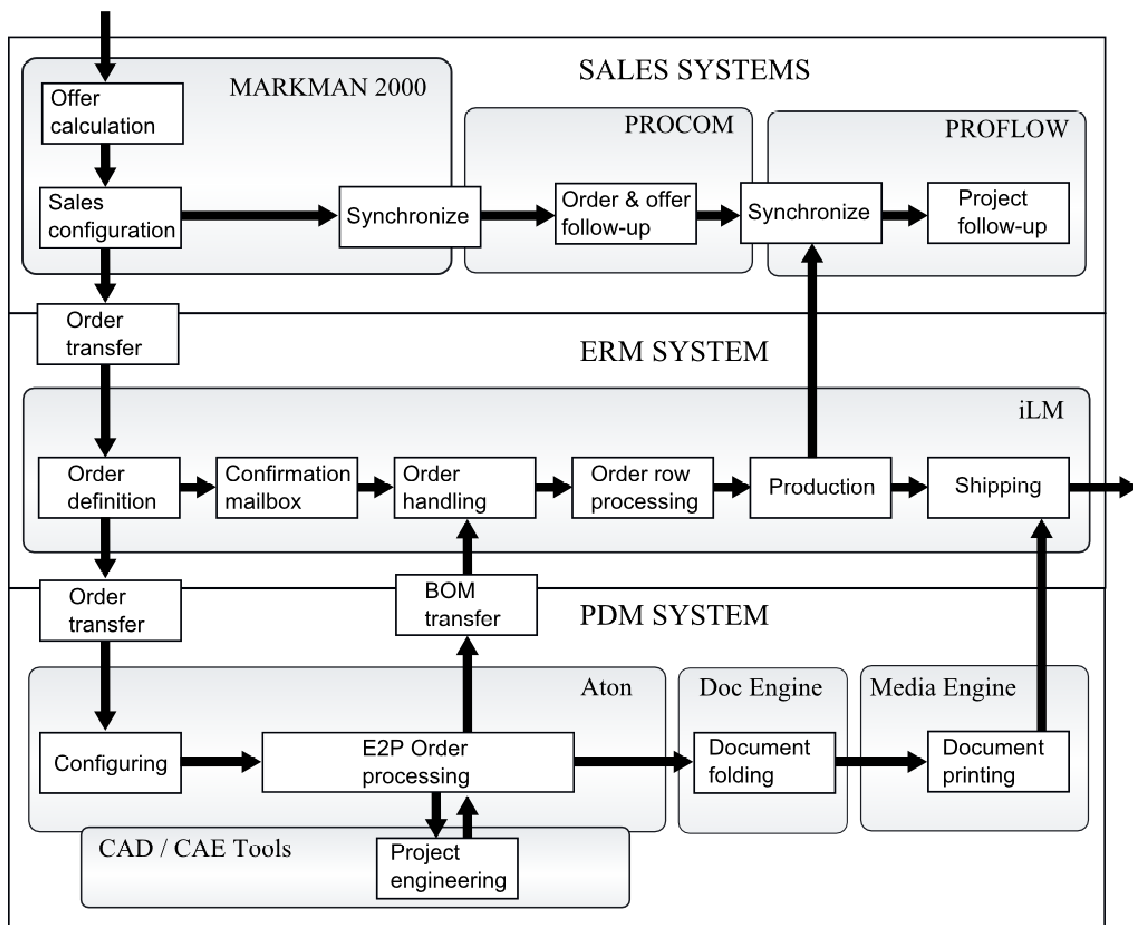


Figure 3. EAI Framework of Konecranes Standard Lifting [1, 2]

2.6.1 Enterprise resource management system

The Standard Lifting Business area uses an Enterprise Resource Management system named iLM. It is an in-house software product and the current version has been in use since 2002. By definition, a full Enterprise Resource Planning (ERP) system should contain a lot of different features that are not all implemented in the iLM. Since iLM

focuses more on the logistics and manufacturing sections of the manufacturing cycle, the term ERM is rather used in this context instead of ERP.

iLM is specifically designed to meet the requirements of the corporation and thus differs in some parts from the conventional ERP system. It is integrated to several other systems in the company, most notably the sales configurator and the PDM system. The sales configurator connects to the iLM using an Electronic Data Interchange (EDI) link to transfer the order information that the iLM later on forwards to the other systems. It also has a connection to several bookkeeping systems, and a project follow-up system that it updates with fresh information once in an hour. [13]

2.6.2 Product data management system

The product data management system is designed to be able to control vast product information entities. In an ideal situation, the system contains all the information that is needed to build a product; product definition and lifecycle information together with the required meta-information to organize all the data. Another important role of the PDM system is to act as a unifying and integrating software platform, a hub, between several other business applications. [14]

The Standard Lifting business area of Konecranes uses a PDM system that is named Aton. One of the most important features of Aton is in its ability to collaborate with the other enterprise systems of the organization. The most important connections to the other systems consists of interfaces to the CAD tools that are used in mechanical and electrical engineering, and an interface to the ERM system used in the production management. Although Aton currently does not have a connection to the sales system, this is one of the things that will be considered when the sales configurator is redesigned.

The mechanical and electrical engineers work with constant changes to existing mass customized products, which emphasizes the importance of item management and fluent version numbering, so that when different departments and people work simultaneously with a crane project, concurrent modifications to the same item cannot take place. This is also the reason why all the engineering tools are tied to work together with PDM system.

All of the CAD tools open their documents directly from the database of the PDM system and save all the modifications to the same system, which also automatically takes care of the management of version history. This ensures that the same documents and files cannot be opened for editing simultaneously in a way that the changes made by

an engineer are overridden by the changes of another. Also the entire revision history and a lifecycle of a single document can be backtracked, if necessary. Aton includes also several features that are designed to control the workflow of changes in the product structure. When an engineer alters the features of an existing product, the changes must be evaluated and accepted before they can be moved into the production.

Another important connection between the enterprise systems exists between the PDM system and the ERM system. The task of the PDM system is to feed the manufacturing specific information, such as product drawings, to the manufacturing platform using the ERM system. Aton communicates with the ERM system by using 15 different order transferring links and the information that is transferred consists mainly of orders and ordered items.

When transferring the order information, the role of the PDM system is to act as a product configurator. When a salesman sells a mass customized product to a customer, the order information contains customer specific unique features as stated earlier. For the ERM system, this information, as it comes from the sales configurator, is useless, unless it is extracted into internal product codes of the company. Aton transfers the sales information structure into internal product codes that are forwarded to the ERM system once the possibly needed variant-specific platform engineering is complete.

2.7 Corporate IT strategy

As stated earlier, the company has had a rapid growth rate recently, and the downside of the corporate acquisitions is that each new company comes with an existing IT platform that very rarely fits into the established infrastructure. Konecranes is addressing the issue of incompatible IT systems by launching a corporate-wide IT strategy that targets to harmonize the IT infrastructure. The IT strategy consists of a thought that all the IT systems in the future should be easily integrated into one another by utilising standardized interfaces. This harmonization is extended also to the look and feel of the different systems throughout the entire organization. The main goal, set by the corporate CEO, is that all the information systems in the different parts of the corporation, and even between the different business areas, ought to be similar.

One of the most notable efforts in the harmonization project is a company-wide business portal that intends to collect all the necessary tools for the different users into a same place. The advantages of having a single portal for all of the web applications are numerous: unified operating logic makes it easier to learn new features, personalized user interfaces can be adapted to the needs of an individual user by hiding unnecessary

information, and all the systems could be made available using a single login procedure. In an ideal situation the user should not even be aware of using different applications inside the portal.

The business portal was launched in the February 2008 and it is based on an IBM WebSphere platform. The strategic guidelines of the corporation emphasize that all the future web applications should be designed in a way that they can be embedded into the business portal. [2, 3]

Another on-going project at the corporation is an implementation of a Global Company Master (GCM) system. A CGM is a centralized system that targets on collecting customer information from various information systems from different parts of the company into one place. This centralized customer information is a part of larger Master Data Management (MDM) entity that aims to collect all the company wide master information into a centralized place. Also an entire Customer Resource Management (CRM) system has been in plans of the company for some time, but currently the actual realization is on a hold. [3, 15]

The corporation also hosts several other IT projects that are not directly related to the enterprise systems but do make a difference in the design process of the new sales configurator. As explained earlier, one of the challenges that need to be explored is a possibility of automatic generation of the sales BOM structure from the information of EBOM structure.

2.7.1 Future of design platform software

Currently the corporation is renewing its old CAD platform with a technology that is purchased from Siemens PLM Software. The new CAD platform is based on a NX and TeamCenter products that were originally developed by UGS Corporation, a company that was acquired by Siemens AG at the end of 2007. NX is the name of a CAD tool of the software suite and it is widely used in engineering industry, being able to work together with TeamCenter Product Lifecycle Management (PLM) software. [9]

One of the biggest advantages of the software suite is the technology used behind it. Currently the CAD drawings that are located in the PDM system cannot be accessed in a convenient manner, since they are all stored in a proprietary file format of the current CAD system manufacturer. The Siemens PLM software suite uses natively JT file format structure that can also be accessed externally. The file format itself actually is a structured 3D database that can be accessed arbitrarily. The file structure forms a tree structure of out of the 3D geometry and Product Manufacturing Information (PMI)

where each tree node points to another 3D file that contains similar meta-information. In an automotive industry example, this would mean that a single file could contain the information of a chassis of a car. The chassis file would again contain information about the engine position and a pointer to another file that describes the engine. Once again, the engine file could contain the information of the engine block and point to another file that describes the injection system. [16]

The tree structure allows the designer to either just pick the engine for editing, or then take out the entire car structure and follow the parts recursively to find the desired part. For a crane manufacturing and sales configurator this would be an advantage, since currently the CAD information that is stored in the PDM system cannot be accessed directly. Different crane parts are stored into the system as separate files and useful information, such as certain distances and diameters cannot be read directly from the system. This leads into a situation where the sales BOM that is used in the sales configurator must be manually updated to adapt the changes in the product structure.

Despite the appealing advantages, the Siemens PLM solution comes with some problems when examined from the point of a sales configurator, since the product drawings that the corporation uses, are already made with older CAD tools. Even though the CAD system is to be changed into a one that is more flexible, the majority of all the product drawings still exist in the old format. Several other corporations have turned their CAD platform from the other systems into Siemens PLM platform at once; Volkswagen AG, for instance, acquired 45 000 TeamCenter licences in February 2008, according to British auto industry headlines. However, the costs of converting all the existing product drawings for a new platform are undoubtedly immense and as of writing this, there are no plans at Konecranes to convert all the old information into a new form. [9, 17]

From the point of the sales configurator, the possibility of accessing different 3D measures directly from the CAD drawings is interesting. The information that resides inside the drawings could be automatically used when generating the sales BOM structure, and in the calculations of the different product variations. Siemens PLM does not, however, offer directly sales configurator solutions, instead the configurator that the company offers is more of a product configurator, which is designed to be used in the handing of manufacturing specific information. [16, 18]

2.7.2 Service oriented architecture meets IBM WebSphere Portal

One of the strategic guidelines for the future implementations of IT systems was the harmonization of the IT infrastructure and the corporate wide business portal to host all

of the future web applications. The business portal is implemented on top of an IBM WebSphere platform, which is also a development platform that is particularly designed for the realisation of Service Oriented Architecture (SOA).

SOA is an IT-architecture that aims to combine different web applications, known as services, into one another seamlessly. The foundation behind SOA lies in the concept of semantic web that was originally presented by inventor of World Wide Web, Mr. Tim Berners-Lee in his book “Weaving the Web” (1999). By definition the term means a network of information that consists of machine-readable smart data instead of regular human consumable web pages that are more common in the world of today. [19, 20]

Today’s Internet works as a medium for people to communicate with each other, and the computers are mainly used as entry points to access all the information. Most of the web content of modern day is only suitable for human consumption, and the knowledge that resides in the network cannot be harnessed automatically. This has led to a current situation where the Internet users are searching and browsing for relevant information using different keyword-based search engines like Google. The semantic web, however, challenges this conventional thinking by defining a set of protocols and methods that are intended to put the information in its context so that it could be used automatically. [19, 20]

A perfect example of semantic web services could be a situation where a user of a calendar application would be booking a business trip abroad. When booking the trip, the calendar application itself would contact the information systems of different airline companies, and present all the possible flights for the user to choose from. Once the user would pick the preferred flight, the application would automatically contact the reservation systems of nearby hotels, check for the reservation statuses of hotel rooms and present a selection of different alternatives for the user. Again, the user would pick a suitable hotel room, and the system would contact the nearby car rental services and present a list of cars that are available for the given date.

All of this time, the user would have just used the same calendar application that is connected to the Internet without knowing anything about the reservation systems and the airline ticket booking systems that have been utilised silently in the background, when the calendar application has been querying several web services using semantic information sharing protocols. This essentially is the idea behind the concept of semantic web and the service oriented architecture. [19]

Service oriented architecture is the technological way to achieve a semantic web. Services in this context refer to the autonomous web services, agents, that are designed

to run independently, and to expose their functionality to the other IT systems. Business enterprises are the first to move towards SOA for the reason that it offers several benefits for coupling enterprise applications together. The benefits include the fact that all the autonomous semantic web services can be implemented in whichever platform that feels most suitable for the purpose, because the standardized description protocols used in information exchange are platform neutral. As the services work autonomously, it is possible to re-use the existing services in other future projects as well, and as long as the interfaces are kept similar, the services in the background can be altered and modified without making any changes into the user interface. [21]

SOA approach makes it also possible to combine different services into one single user application. From the standing point of the EAI this is a clear benefit because most of the big enterprise systems tend to have information that is related to other information residing in another system. With a sophisticated SOA approach it is not only possible to combine different information from the different sources, but it is also possible to do it automatically on a single sign-on basis. Image 4 shows how a single sales configurator service could be reused in several end applications at the business portal and also how the other services could be used to extend the usability of an application. [22]

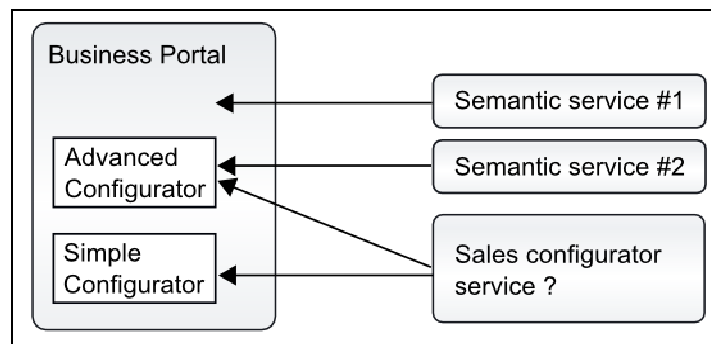


Figure 4. Possible SOA Architecture reusing services

WebSphere Portal by IBM is one of the first commercial server software products that are initially designed and built to support the SOA architecture. Business portal of Konecranes is designed to act as a central hub for several different web applications of the company, and it is something that needs to be taken into account when designing the new sales configurator. Although many of the requirements posed for new sales configurator point to a direction that the new system ought to have a possibility for offline use, several other aspects of the project point to integrating the configurator into the business portal using SOA architecture, such as the requirement of connectivity with other information systems. The in-depth requirements for the new system are presented in chapter four. [22]

3 CURRENT SALES SYSTEMS

This chapter begins with a short review of how sales configurators are dealt in the scientific literature. As most of the sales configurator implementations are case specific, the actual definition of a sales configurator and its responsibilities vary much. The purpose of the literature review is to introduce several different viewpoints on how sales configurators are perceived on the whole.

In addition to the literature review, this chapter describes the current sales systems of the corporation together with their functionality. Before the sales configurator can be designed even on a high abstraction level, the functionality of the previous configurator must be fully understood. Additionally, a big part of the old features are also requested from the new system. This chapter presents the functionality of a current sales system in its usage context. The usage context principally covers a set of different sales processes that are used in the company to describe and manage the actual sales event.

The usage context, use cases and sales processes are depicted using Unified Modelling Language (UML) charts. The full understanding of sales processes serves the purpose of understanding the usage environment and the use cases of the sales configurator. For the new sales configurator project to succeed, it should at minimum contain most of the features of the old one, and thus all the use cases and sales processes will be used to assist the architectural outlining of the new version.

3.1 Sales configurators in literature

Scientific articles written about the sales configurators are few and far between. Since the sales configurators for engineering-to-order production are expert systems, whose idea is to present the knowledge of the entire product structure in a manner that is easily understood by a customer, articles covering artificial intelligence from the point of the sales configurators are preferred over the ones that merely treat the sales configurators as electronic price catalogues. Some of the articles are written about designing a customer interface for a product configurator, which fundamentally is the same thing as a sales configurator.

3.1.1 General approach towards configuration problem

Bramham and Maccarthy approach the implementation of configurators from a business-oriented point of view. The implementation of a new configurator system is

seen as a mean to increase the customer value of a product. More specifically, the configurator is viewed as a tool to close the gap between the individual customer needs and the capability of the supply chain network, by exposing the diversity of the product range in a manner that can be understood by the customer. The problem involved in the engineering-to-order and configure-to-order production approaches, is how the product information is presented to a customer so that the true benefits of the diversity that the modular product structure offers can be harnessed to a full extent. [23]

The greater cause behind the implementation of the configurator systems is that they are seen as tools that make it possible for the entire supply chain to operate in a more demand-driven way. Not only do the configurators help to expose the product structure to the customer, but they also serve as important tools to verify the validity of a product variant, and to extract all the information from the customer that is needed to build a proper product. [23]

The configurators have been generally accepted as powerful tools to capture customers' requirements, and ever since the 1970s the research of the configurators has been on the rise. The configurators are traditionally applied to the computer industry where they have been used to configure combinations of computer hardware, such as R1/XCON made by Digital Equipment Corporation (DEC) for VAX computer systems already back in 1978. Ever since, a large number of configuration expert systems were developed for the purposes of the computer industry, such as Cossack for Xerox PC or MICON for the configuration of single-board computers. The advantages of configurator software have been also noticed in the traditional manufacturing industry, most notably the areas that operate by the means of mass customization, and today the configurators are applied to various different fields of businesses, ranging from the automobile industry to interior design applications. [7, 24]

The sales configurators are often portrayed as expert systems, since they are systems that incorporate information of human origin, and use the knowledge to provide problem analysis to the users of the software. Expert systems are commonly researched topic in the information science and the approaches used in the design of such systems come in many flavours. Many sales configurators use a conventional rule-based reasoning logic to harness the knowledge. Rule-based reasoning basically signifies a problem-solving method that is based on chaining a set of IF-THEN –rules. These rules are run against the knowledge base thus providing a desirable result. Wang et Tseng point out in their science article that these kind of systems often suffer from the maintenance due to the lack of separation between domain knowledge, control strategy and the spread of information on a particular attribute over several rules, when the configurator system is complex. [24, 25]

One of the most cited articles about sales configurators is one that is written by Mittal and Frayman. Their article was the first one to treat the configuration task as a constraint satisfaction problem instead of a traditional forward chaining rule-based reasoning problem. Constraint satisfaction problems are mathematical problems where one must find those states that satisfy a number of constraints or criteria. A traditional example of a constraint satisfaction problem is how to place eight queens on a chessboard in such a way that none of them attacks each other. The article aims towards developing a generic model for configuration task that can be used in various different applications and offers a number of methods for dealing with the problem in such environment. [26]

The generic configuration task is defined using a minimal number of assumptions in such a manner that it consists of components that can only be connected to other components in fixed and predefined ways, in conjunction with the idea that the components cannot be modified to obtain arbitrarily connectivity. Every component is considered to have ports that are used to connect the components with each other, and each component locally describes the constraints on the other components that can be connected at each of its ports. Thus the presented solution does not only describe the components but also how to connect them together, which keeps the solution model general enough to cover a broad array of different configuration problems. Customer requirements in the article are referred to as functions, since the approach is to find a solution that fulfils the original requirements by providing a required functionality. The resulting solution integrates a functional model into the configuration model by defining a mapping from functions to key components, which must be part of the configuration if the function is provided, thus forming a functional architecture. The functional architecture assists also in the acquisition of customers' needs, because in most cases the customers are not interested in the detailed product topology but rather like to specify a set of functions that the product must offer. [26]

Minimal number of assumptions in the generic approach leads to a very complex problem, thus the authors introduce two restrictions to the model. The first restriction is based on the observation that most of the artefacts are always designed with some purpose in mind, which limits the problem domain. When the functional architecture behind the configured product is well known, it is unnecessary to compute each product variation and instead one can focus on those possibilities that match the pre-defined architecture. The second restriction bases on the thought that it is possible to identify some particular components of the product design which are crucial when implementing some required functions. The second restriction introduces a concept of a key component, a component that is more important than the other components because it limits the usage of other functions and fulfils a function by itself. [26]

The two assumptions, functional architecture that is used to denote customer requirement, and a key component approach fit together quite well. The functional architecture allows the project designer to decompose an artefact along the guidelines set up by the original design architecture, and the key components can be mapped to the architecture to behave as “planning islands” allowing the required functions to be configured somewhat independently. [26]

Figure 5 illustrates a difference between a traditional rule-based reasoning system and a system that uses the constraint satisfaction approach. The left side of the figure shows a situation where all the components are treated in an equal manner using a rule-based logic eventually forming a decision tree. The right side of the figure approaches the problem by defining certain key components superior to the other components, thus implementing the functional architecture. The components have ports to the other components that can be connected only if both of the ports are compatible with each other. The components may also be interdependent of each other.

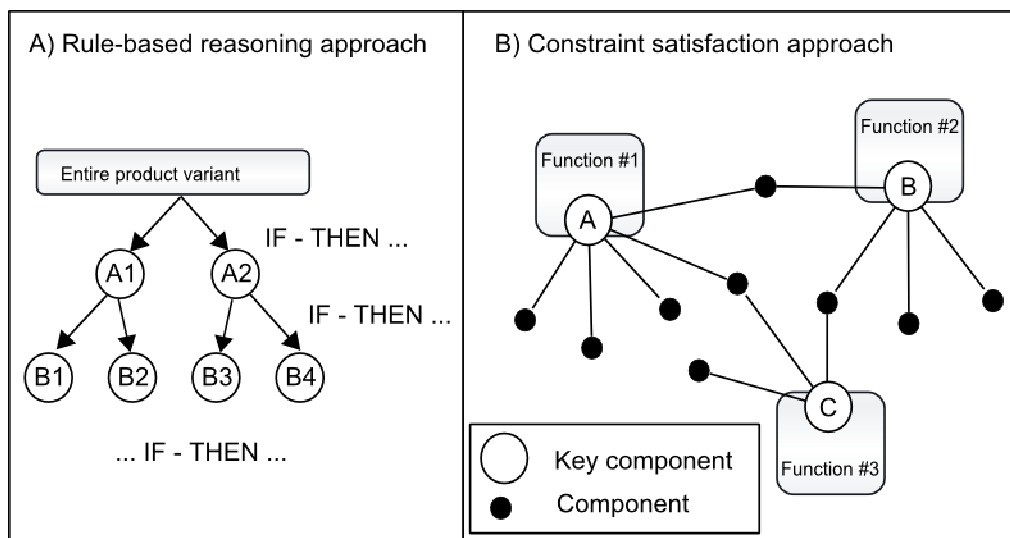


Figure 5. Configuration of a product using (a) rule-based reasoning (b) constraint satisfaction approach

Wang and Tseng take the concept of constraint satisfaction problem further by introducing a statistical approach to improve the efficiency of configurators in their science article. To approach the configuration problem a Bayesian probability network is deployed to the component structure to reflect the physical structure of the product. A probability-based configurator is foremost designed to deal with the task of configuration under uncertainty, and to capture the customers’ needs quickly with less information available. [24]

The idea behind using a Bayesian probability network is to sequentially present the component with the most relevant information to a customer to limit the upcoming selections in the pool of available components. Basically, the customer selects the most informative component to be specified first and then moves forward to the next most informative component. The term “most informative component” is used to refer to a component that contains the most relevant information about the customers’ special interests. The article suggests that the knowledge for information value of a component could be gathered from the experts working at the same field by forming an acyclic probability graph with probability information assigned to each node. The advantages of this solution are seen as steering the configuration problem more in to a customer-orientated direction by eliminating the uncertainty from a decision making process and by forcing to take the customers’ opinion into account when the physical structure of a product family is designed. [24]

3.1.2 Implementation of configurator systems

Jiao and Helander cover the entire development cycle of an electronic configure-to-order platform for the customized product development over the Internet. The authors approach the configurator development through a case example of a company manufacturing injection-molded parts. The system aims to integrate a number of web-based services into a collaborative web of interactive commerce. The implementation of the configure-to-order platform is justified by an increasing demand for customized products, and the possibilities that a configurator offers in identifying customers’ needs. [27]

The Internet-based software solution was chosen as a primary development platform for a number of reasons: it offers scalability, a worldwide collaboration network, and compatibility across the diverse information technology platforms. An Internet-based solution is also considered viable for the reason that the case example company operates in several geographical locations and the product development requires negotiations and collaboration between several different departments. Also, a number of CAD software vendors have selected similar and compatible technologies, which affected the selection of the platform. [27]

The injection-molding company consists of several different departments, such as marketing department, design department, engineering analysis department and a manufacturing department. The customers mainly interact with the sales and marketing department and the transformation of initial customers’ needs into final manufacturing information requires lengthy process of collaboration between the several different departments. The problem of varying customer requirements is answered by designing

product families that also help to reduce the overall development costs. The product family information is stored in a separate knowledge repository and the repository per se is organized around a unified GPF (Generic Product Family) master model. [27]

The authors consider the GPF model to be the kernel of the platform-based product customization. It is being described as “an information model that encodes not only the mapping relationships between product functional specifications and the corresponding design solutions, but also the design and manufacturing decision-making criteria that must be fulfilled for the applicability of each design solution“. Briefly, the GPF information model is a knowledge database that contains a generic product model, which is used during the mass customisation phase in the creation of product variants. It does not only capture the decision-making criteria within each individual design domain, but it also contains mapping mechanisms that are utilised when feasibility of an individual product variant is assessed. Furthermore, it features the functionality to map the information between several different design domains so that the same information model can be shared among the different user groups and purposes. [27]

The system architecture chosen for the platform follows a common three-tier architecture and it is depicted in Figure 6. It utilises JDBC/ODBC (Java Database Connectivity / Open Database Connectivity) techniques for the integration and it consists of a number of clients, database servers and application servers. [27]

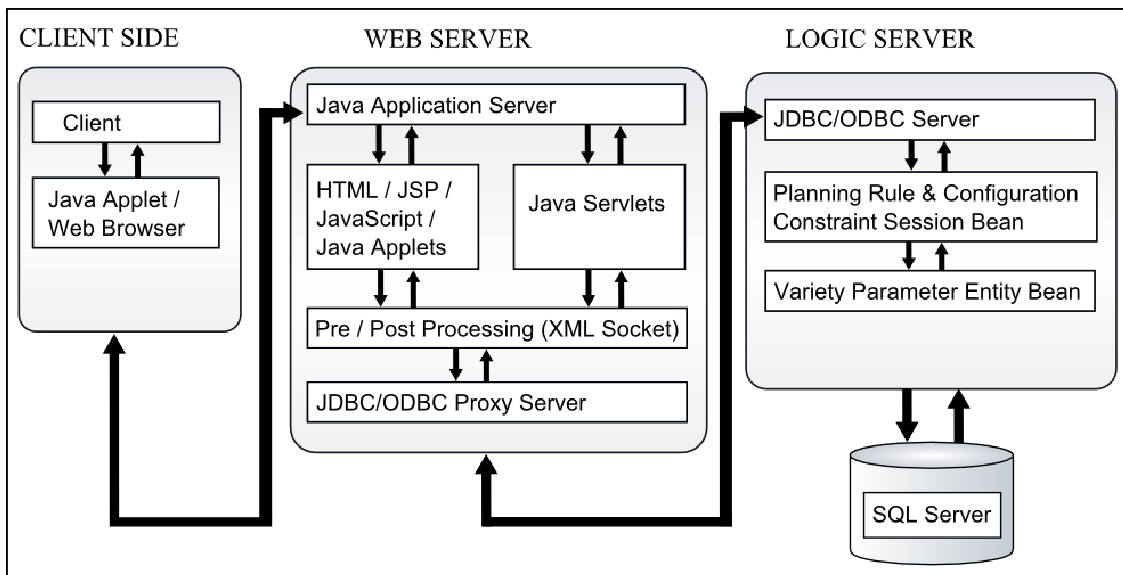


Figure 6. System architecture based on J2EE (Java 2 Enterprise Edition) [27]

The system architecture is designed to be consistent with a multilayer distribution application model of Java 2 Enterprise Edition Platform. The selected architecture makes it possible to incorporate several different techniques, such as Java Server Pages

and servlets for the implementation of server side logic. The client part of the architecture is responsible of the user interface, and its main responsibility is to translate the tasks and results to some form that the user can understand. In this case the client side segment needs to support many different users that are involved in the order-to-production cycle. The user interfaces are implemented using various Java-based techniques, for example running Applets through a web browser. The web server layer feeds the client layer of the application by providing all the necessary dynamic content that the client layer outputs to the user. The logic server coordinates the application, processes the commands and makes the needed logical decisions. Its responsibility is to transport and process the data between the two surrounding layers and it could be considered as the main part of the application for it handles all of the logical functionality. SQL (Structured Query Language) server contains the database of persistent information and most essentially the GPF master model, which is the heart of the application. [27]

Other technologies that were chosen for the configurator application consists of API (Application Programming Interface) libraries and adapter technologies that aim to help in the development and to provide a more convenient way of linking the configurator system into other existing legacy systems, such as the PDM and the ERP systems. An XML (Extensible Markup Language) parser library XML4C by IBM was selected to the project to assist in the data exchange between the GPF model and the other parts of the application. An Aglet library, on the other hand, was chosen for the development of mobile agents that are used to encapsulate connections to the other existing information systems. The Aglets act as templates that adapt to the contents and formats of specific information exchange on the input side, and output the relationships to the legacy systems. An Aglet generates a variable that provides an encapsulated interface to a legacy system thus making it easy to plug the legacy systems into the configurator application. [27]

3.1.3 Implementation of knowledge base

One of the problems that the configurators face is an ever-growing complexity of the underlying knowledge base, and the task of keeping it updated and maintained throughout the lifespan of the sales configurator. Felferning et al propose method for building knowledge bases out of generic product models using graphical conceptual modeling and employ a standard design language (Unified Modeling Language, UML) for modeling of configuration databases. [28]

Keeping the generic product model up to date and acquiring knowledge for the knowledge base is a complicated task often requiring the database engineer to be an

expert on the actual database implementation as well as in the product knowledge. A standardised graphical description language for building knowledge bases can be used to reduce the complexity to the level that the actual product knowledge can be acquired without any knowledge of programming. Principally, UML is a conceptual modeling language, which is widely applied in the industrial software development processes and it can be used to describe the architectures of several different application domains. UML originates from the software engineering industry but it has been lately adapted to other industrial development processes as well. Architecture description languages provide basic concepts for modeling complex entities in a way that the model is detailed enough to be used automatically, but offer high enough abstraction level to make the handling of information easier. Furthermore, the UML language offers possibilities for having separate views to the model and for extending the language, to meet more user-specific needs, whereas the built-in accuracy of the presentation model can be harnessed into automatic validation and verification of the given design model. Model-based diagnosis techniques are already old research topics and they were initially developed to identify faults in physical devices, such as circuit boards, but later on, these techniques were adapted to the field of computer science for diagnosis and debugging of software systems. The established methods can be used to identify incorrect clauses and constraint violations by utilising the information about the expected and unexpected query results. The possibility of having several different views to the product structure also makes it possible to utilise the functional architecture-based modeling concepts such as the one proposed by Mittal and Frayman. [28]

Felferning et al propose the configuration environment to consist of three different elements as depicted in Figure 7: knowledge acquisition, configuration and reconfiguration. The knowledge acquisition focuses on building the actual generic product structure by using the UML notation and the resulting models are automatically translated into a logical representation executable by a configuration engine. Once the configuration model is created and translated, the knowledge base needs to be validated, which can be done automatically by using model-based diagnosis techniques. The validation can be done by using positive and negative configuration examples and the diagnosis methods may also provide information of the parts of the knowledge base that are causing unexpected behaviour. Eventually the outcome of the validation phase is a set of logical sentences that can be mapped against the generic product structure allowing the problems to be assessed and repaired in a graphical environment. [28]

The actual configuration phase of a single product is rather trivial, once the rule sets already exist in a hierarchical and executable product model. Once the user requirements are collected they can be run against the knowledge base using various different methods for optimisation, such as a breadth-first method for fast error

checking or a branch-and-bound optimisation algorithm for finding the most optimal solution. Presented framework enables a reconfiguration of the existing configurations in a way that new customer requirements can be gathered and run against the knowledge base. [28]

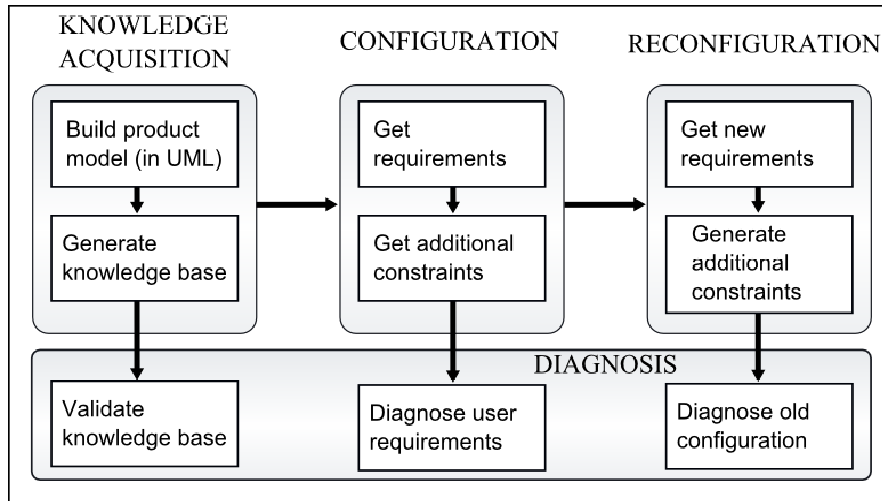


Figure 7. Configuration environment after Felferning et al [28]

The authors present an UML based configuration example of a desktop computer as a case example. The knowledge base depicts the generic product structure that represents all the possible variants of a product, whereas a concrete configuration, the result of a configuration process, represents an instance of a generic product schema. The UML modeling of knowledge base begins by defining a profile for the UML chart, and it is used to specify a special set of classes and constraints for the given task. The configuration models are defined using the component-port presentation to ease up the translation from an UML chart into an executable rule set. The component-port presentation is a well-established method for representing and solving configuration problems and it was already used the functional architecture that was presented earlier.

A configuration model example is shown in Figure 8 and it consists of several different types of objects:

- *Component types* that represent the parts that the final product can be build of. The components are characterized by attributes that have a predefined domain of possible values.
- *Function types* that are used to model the functional architecture of an artefact and are also characterized by attributes.
- *Resources.* Part of the configuration problem is seen as a resource-balancing task, where some of the components produce some resource that the other components consume.

- *Generalization types* are used to group the components with similar functionality into a hierarchical structure. Generalization types are not actual components and when reading the diagram they ought to be considered more as if they were headers of a certain group. In example, a server-os in a diagram is a headline for either server-os-1 or server-os-2.
- *Aggregations* between the components indicate that a component consists of several subcomponents. In example, a motherboard contains a sub part called cpu.
- *Connections and ports* depict how the different components are interconnected to each other, thus forming a graph. The ports contain knowledge about different parts that can be connected to each other.
- *Compatibility relations* are restrictions that limit the selection of components. Some components cannot be used together in the same configuration because they are incompatible with each other. Additionally the existence of one component may require the existence of another special component.
- In addition, some constraints to the product model cannot be expressed graphically and thus they are formulated using Object Constraint Language (OCL), which is an integral part of UML definition.

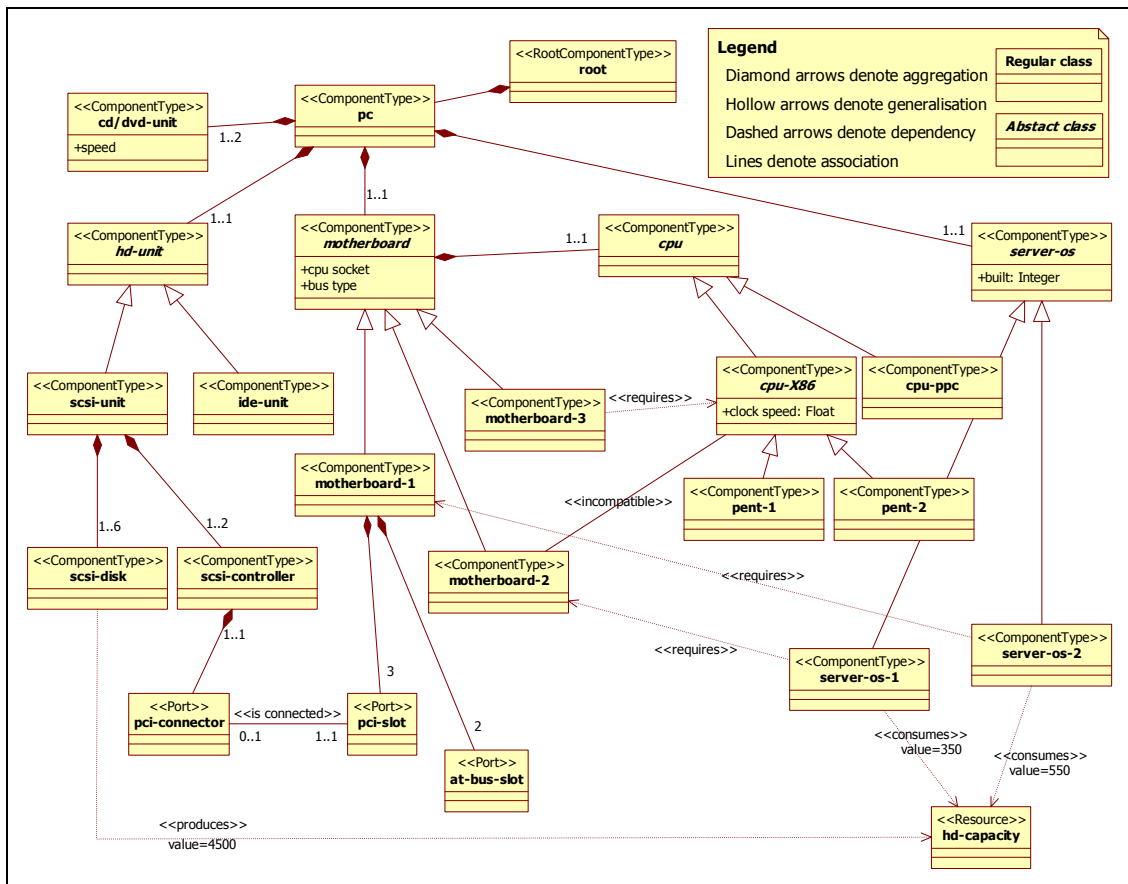


Figure 8. Conceptual product model of PC computer freely after Felfering et al. [28]

The definition of UML offers several built-in mechanisms for structuring the diagrams to keep them simple even with larger and more complex models. One of the mechanisms is the possibility of having several different views (graphical depictions) on the underlying model, which can be used to build distinct views for functional, connection-oriented or a certain substructure of the product. The definition also includes a built-in packaging mechanism making it possible to partition the model into smaller substructures for easier editing. [28]

Because the original configuration phase was dealt by compiling the knowledge base into an executable set of rules, the same configuration engine and the optimisation routines can be harnessed into reconfiguration part of the project as well. The main idea behind having an executable rule set is that the configuration parameters are collected from the user and run against the configuration rules using a separate configuration engine. Since the rule set is generated from an UML chart that uses graphical notation, adding new features and updating the model does not require the updating person to be a programming professional. The configuration engine itself functions in a similar manner as a programming language interpreter in the computer science: it reads the rule set that is compiled in a convenient format and runs the collected configuration parameters through the engine, the output being an optimised product that satisfies all the conditions of the product model. In a case that the configuration fails because of missing or inconsistent information, the engine is able to clearly identify those components and properties that causes failing and even graphically highlight them in the user interface of the configurator. [28]

Reconfiguration support is an important thing to be considered since various applications have an increasing demand for after sales support. Demand for the reconfiguration of product individuals rise when a customer wants to upgrade the existing system to provide new or altered functionality, or when the parts of the existing system are broken and have to be replaced by a newer version of the component for the original components are no longer available. The goal of the reconfiguration process is to compute a configuration, where the most parts of the existing configuration can be preserved so that the number of needed changes is minimized. Having an executable rule set assists in achieving the goal, since the reconfiguration process can be guided by optimisation functions to find the minimal changes in the configuration or the product structure so that given the configuration passes the configuration engine. The other optimisation functions may include assigning the changes in the configuration with different costs, in example for the situations when a change in the parameters may be cheaper than exchanging an entire component. [28]

The model presented above responds to the ever-increasing knowledge acquisition problem with complex products. With the use of UML charts, the editing of a product structure could be done without exhaustive understanding of computer programming, and in the knowledge acquisition phase, any tool supporting the UML could be used as long as the resulting UML models could be stored in an XML representation. It also incorporates the functional architecture and offers several views to the product model, which are commonly utilised methods in the configurator design. [28]

3.2 Current sales configurator

The current sales configurator that the company uses is called Markman 2000, and it was originally developed more than twenty years ago, although redesigned in 1997. The history of current Markman now spans over a decade and during this time the project has more than doubled its original size and became so complex that keeping the program updated and maintained requires tremendous efforts. In the terms of software engineering, the project has become bloated. The application is originally written with Visual Basic versions 5 and 6, and it consists of 32 different calculation components compiled as DLL (Dynamic-link Library) files, more than ten different internal databases and a control module, which acts as the main application. Altogether Markman has more than 1.5 million lines of program code scattered in various DLL files and the control module, making it a very large project to be written Visual Basic. Markman is a standalone application that does not need a network connection for crane configuration. It is, however, able to synchronize information, transfer the orders to the production and utilise an external server for generating a 3D model out of the configured crane if a network connection is present.

Visual Basic itself is a Rapid Application Development (RAD) programming language that is designed for quick prototyping of Graphical User Interface (GUI) applications in a Microsoft Windows environment. Visual Basic, however, is currently deprecated in favour of .NET programming language families, and the extended support by Microsoft ended in the March of 2008. It still has a very large user base in the industry and it competes with other programming languages such as Java or C++. Nonetheless, Visual Basic is most of all designed to be a RAD language, meaning that the main emphasis on the Visual Basic development has traditionally been getting visible results fast, not the structure of the code. It also means that the Visual Basic projects are never intended to be very large and easily maintainable, which is why some software companies have had a policy of rapidly prototyping their projects using Visual Basic, and once the prototype has appeared feasible enough, the entire application is rewritten using some other programming language that is easier to maintain, usually one that supports object

oriented methods to full extent. Another popular method has been to design the user interface with Visual Basic and then rewrite the DLL components with some other programming language. [29, 30, 31]

Rapid application prototyping usually comes with compromises regarding the execution speed or features. Visual Basic has been able to compile native binaries starting from the fifth version that was published in 1997. This improvement in the computation speed reduced the necessity of converting the Visual Basic code into other languages. Sufficient calculation speed was also enough for Markman, and thus the computation speed in crane configuration never required the project to be rewritten in another language, and Moore's law, which states that the computer performance doubles every 24 months, has taken care of the calculation speed even though the project has increased in complexity during the years. [32, 33]

The speedup never solved the problem that Visual Basic projects are not designed to be easily maintainable, and after ten years the project cries to be rewritten. During the time of Markman's existence, the entire computer industry has taken a huge leap forward and the world has seen things such as the rise of high speed Internet, virtualization of desktops and a rapid growth of server-side web applications that are nowadays able to handle complicated tasks which traditionally could be done only using standalone applications. High speed Internet has also brought up number of undesired features such as outbursts of computer viruses and malware, which has enforced the IT departments to limit the user privileges of workstations. Currently Markman is unable to operate on a workstation with the limited user rights since is not initially designed to do so. Being a standalone application, Markman also requires regular updates to keep the pricing and product information updated, which poses problems when working in limited rights environment since the installation of upgrades require full administrator privileges.

Other reasons that emphasize the need of a complete redesign lie in the fact that the entire product structure in the company has changed much during the years, and since Markman has been performing in a honourable way, several product families that were not initially included in the system, have been added afterwards. This poses a problem, since the fixed user interface of Markman was not designed to adapt to very large-scale changes and every update requires a considerable programming effort. It also requires the programmers to be crane specialists, since adding a new product family incorporates programming the new product knowledge inside the system. The fact that Markman was not initially designed to handle such a large group of different crane models has also lead to bloating of the user interface as the number of different possible component selections has increased dramatically. The user interface has eventually become more of an engineering tool than a sales tool and the new users find it very difficult to learn. [1]

Even though Markman currently comes with many apparent flaws, most of which exist mainly because its role has changed from the application that it initially was, it still has maintained a relatively high user appreciation. Despite the complicated user interface, many older salesmen and the advanced users find it pleasing that they are able to have so many different selections and features to configure the crane in exactly the way they want. Another acclaimed feature is that Markman is able to configure a crane even without a network connection. Many of the crane deals are made in the developing markets and industrial sites where the availability of network connections cannot be taken as granted, and according to the words of a key user, the salesmen have reported that the fact that Markman that is able to function without a network connection has given a clear market advantage over the competitors in several cases. [34]

3.2.1 Architecture of current sales configurator

The architecture of Markman is designed so that the main application called ‘control module’ contains the user interface and also wraps the rest of the application together. The crane specific optimization and computation are distributed to 32 different calculation modules that are compiled as DLL libraries, which are called from within the main application. The persistent crane information is stored in 13 different databases that contain information about the different options, prices and other crane and brand specific data.

The crane calculation modules are compiled as DLL libraries, which is Microsoft’s way of creating shared software libraries. Virtually every DLL file is an application that provides certain functions that can be called from external applications using a COM (Component Object Model) interface, and they can be shared among the different unrelated applications. The DLL libraries come in two different flavours: the conventional versions, and the ActiveX kind. The conventional version of a DLL library can be called from within almost any application without any special requirements, and as long as the DLL file is found from the system, the functions that reside inside the library can be called using static or dynamic runtime linking. The ActiveX DLLs, on the other hand, come with some special requirements, as they were initially designed to contain an entire application that could be embedded into another application. The idea behind the concept of an ActiveX library is that one application can be embedded into another application seamlessly with very little effort by using standard interfaces, and because of the very interfaces they cannot be referenced directly, instead the DLL libraries must be registered into operating system that eventually provides the interface for calling the application.

The calculation modules of Markman are mainly using the ActiveX flavour of DLL libraries, since most the calculation components are built using Visual Basic 6, which does not even offer a possibility to compile regular DLL libraries by default. Some external libraries that are licensed from third parties are using the conventional DLL architecture but none of these are regarded as being calculation components. All of the calculation components and their responsibilities are listed in Appendix 1. The usage of the ActiveX components should be taken into consideration because the corporation has expressed their willingness to re-use the existing calculation libraries in the new configurator as well, and not all of the programming frameworks are able to handle calling methods of ActiveX libraries. This may limit the selection of available and feasible technology platforms, since Java JNI, for example, is unable to reference ActiveX DLLs using the standard interface. The other obvious limitations rise from the fact that the DLL libraries can only run on the Ms Windows operating system. The complete requirement analysis with more limitations is presented in the next chapter.

Persistent information in Markman is stored in several locally stored Ms Access databases that are part of the Markman installation. The databases contain all the generic product models, pricing information and user information. The Ms Access databases were chosen at the time when there were no other feasible alternatives for lightweight databases that run on a desktop environment, and they offered a relative compatibility with the SQL statements. However, internal limitations of the database format have forced the database to be split into several smaller databases. A distinct feature in the database implementation is that the different brand specific installation packages of Markman contain different sets of databases, due to the fact that different brands ought not be able to access the information of each other for business reasons. Access does not offer strong enough encryption so that the same information package could be safely delivered to all the different users without risking revealing restricted information to competitive brands, and building a separate brand specific installation packages, that do not contain any unnecessary information, solves the problem.

The installation packages are delivered to end-users by using an Internet update mechanism. The mechanism provides a semi-automatic update system that serves the user with the most current version of configurator, and determines an appropriate installation package based on the user's privileges. The actual installation, however, must be done manually once the installation package has been downloaded, and in order to enforce the users to keep the system updated, Markman has a built-in timer system that prevents it from running after a new version is released and the old version is outdated.

Other notable feature is Markman's ability to connect to an external drawing automation server (DAS) after the configuration process. All of the configured parameters are sent to an external server that computes 3D model drawings out of configuration information. DAS application itself is a product of a CAD manufacturer Vertex Systems, but the automation software that steers the system and the server logic is built specially for Markman. The 3D models that are generated from the configuration information are accurate enough to be used in the manufacturing of a crane, but they are also used for the sales purposes when presenting a newly configured crane variant for the customer. Installation package of Markman comes with a lightweight Vertex viewer application that is able to display the DAS models on screen. An example rendering of DAS model is shown in Figure 9.

DAS images are not used exclusively for eye-candy and manufacturing purposes either. When a crane model requested by a customer contains so special features that it cannot be configured automatically, a template crane close to the required crane is designed and drawn with the help of Markman and DAS server, and the mechanical engineers implement the case specific features on top of the created template model. In a sense, the DAS system handles the mechanical engineering automatically for those cranes that can be automatically configured.

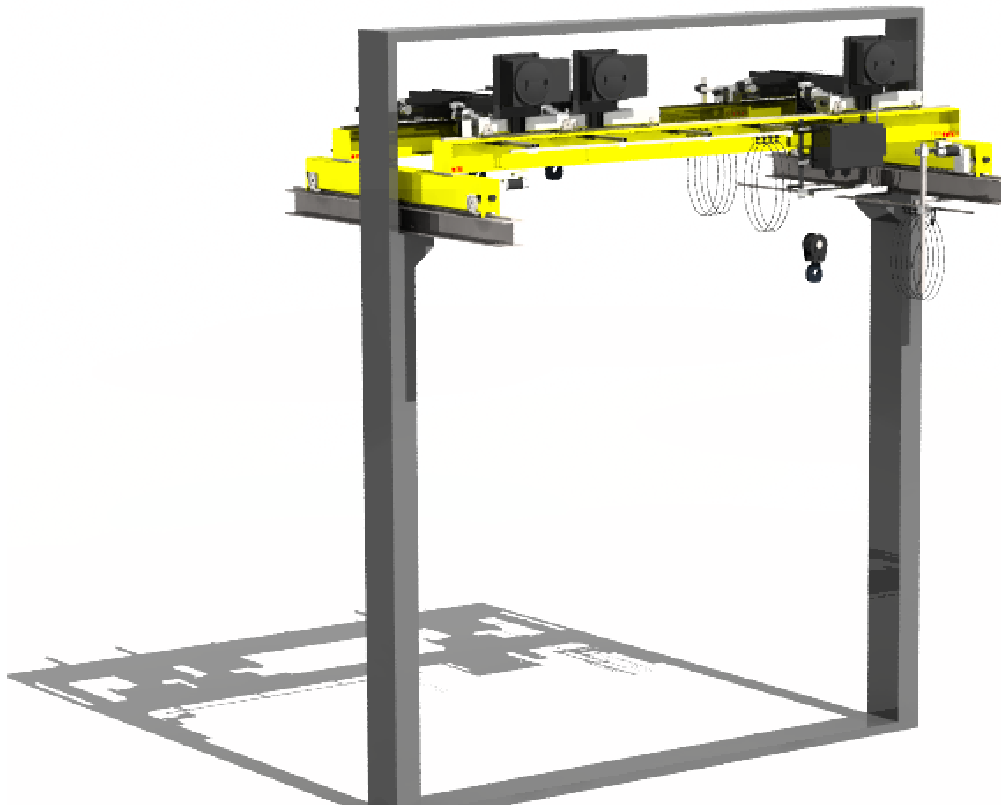


Figure 9. 3D rendering of a crane model created from DAS drawing.

Markman also offers a possibility for automatic drawing generation without a network connection using a subsystem called CadMan, which it is able to generate 2D drawings out of a configured crane variant. The use of CadMan has been decreasing ever since the introduction of 3D drawing automation system, although, CadMan is still used often when salesmen are sketching out the measures of a crane variant for the reason that CadMan is able to create drawings with proper measurements instantly, when creating a 3D image with DAS server requires several minutes of calculation.

When a configuration is complete, Markman bundles all the configuration information together with the CAD drawings into a single package that is sent to a common order-handling mailbox of the sales department, who later on strip the order apart and handle the ordering of required components. Markman is also able to send single component orders directly to the ERM system using EDI format but the orders containing complete cranes need to travel through the sales department before entering production.

3.2.2 User groups and use cases of current sales configurator

Markman has several different user groups and use cases in addition to the obvious group of salesmen calculating the offers and configuring the cranes. The different user groups that relate to Markman are presented in Appendix 2 in a form of a UML actor diagram and their relation to different use cases are depicted in Appendix 3. The user group diagram should be read in a manner that the user groups that are written in italics depict a generalization of a user group and should be interpreted practically as a header for those user groups that are connected to it. As an example ‘Sales department’-group should be understood to consist of a crane sales department and a component sales department.

The only user group that interacts directly with the regular end-customers is the frontline salesmen that may either be a component salesman or a crane salesman. The differences between the two different frontline salesmen groups reside in their general responsibilities in regard to the products that they are selling. The component salesmen sell crane components for the customers that wish to upgrade their old crane configuration or have needs for spare parts, whereas the crane salesmen sell the entire crane configurations. What separates the two different frontline salesman groups may not be apparent in all the cases, as in some business regions a single salesman may very well represent both of the groups. The use cases of frontline salesmen are relatively similar to each other and the both of the user groups use Markman to calculate offers to the customers and to create offer letters out of the calculations. Both of the user groups are able to create EDI orders, which means component orders that are transferred directly to the ERM system of production platform, and they both also synchronize their

offers to an external offer manager system, ProCom. In addition to the component orders, the crane salesmen are able to create entire crane orders, bundled in ZIP packages that are transferred to the competence center for order handling, and they also utilise the DAS server to generate 3D crane drawings for customer's acceptance. [34]

The term regular customer was used to depict a customer that represents an industrial entity or another third party customer who wishes to purchase a crane or to modernize the existing one in a conventional manner. However, the customers also come in many flavours and some of the customers use the configurator themselves to make their own purchases. These customers are either component customers or licensors who use the sales configurator for their own purchases. The licensors configure the entire cranes much in the same way that the frontline salesmen do, but instead of working for the Konecranes Corporation, they represent other independent crane companies that sell the cranes purchased from Konecranes to their own end-customers. The component customers, on the other hand, purchase the crane components for several different usage purposes. They may represent independent crane manufacturers or they can even be hardware stores that sell spare-parts for smaller cranes and chain hoists. [34]

Competence center represents the part of the corporation where the crane technology related knowledge is gathered. Their responsibilities are to oversee and consult the other departments in crane knowledge related questions, and they are also the user group that receives the component and crane orders from the frontline salesmen. Competence center does the order handling, chop the complex orders apart, and in the end they also steer the application design, whenever special features that cannot be calculated with a configurator, are requested. They cooperate closely with the sales and offer design departments, and to some extent the competence center could be regarded as a sales department of the future where all the crane knowledge is gathered in one place. Also, the key users, who administrate Markman system and its user accounts, work closely together with the competence center. [34]

In a regular case, when the crane orders can be configured easily, the order handler can simply open up an order file that has come from the frontline salesmen using Markman, and accept it into the production by feeding the order information to the ERM system. Then again, even if the crane cannot be fully configured, Markman is still used to craft an offer and this is the situation where the sales department and the offer design department steps in. The sales department acts as a middleman between the offer design and the frontline salesmen. The offer design department configures a template crane using Markman, utilise the DAS server to compute a base drawing for it, and finally add the required specialities that cannot be configured automatically. The sales department

evaluates the project; sets prices for the specialities and in the end, make a decision if the project is financially feasible enough to carry out. [34]

Another part of sales department deals with the component sales and is also in charge of crane modernization offers. When an old crane is to be modernized, it may originally be a product of another crane company, or at least it is most likely so old that it cannot be reconfigured using Markman. The component sales department acts in the same way as the crane sales department in determining prices for a project, and making a decision whether the project is financially worth continuing. Nonetheless, the component sales department uses Markman to estimate the costs of different components that are required for the crane project, by picking up individual components that are needed for the modernization. So it is important that the configurator does not only configure the entire cranes, but also offers a possibility to select the individual components and calculate prices for such selections. [34]

Other user groups that are tied to the context of the sales configurator involve the crane factories and the platform users. The crane factories eventually receive crane orders and manufacturing drawings. In some cases when the drawings are not yet made, the crane factory uses Markman to print out the DAS drawings for manufacturing purposes, and even in those cases when the drawings are provided, the orders are reviewed and recalculated for safety reasons. The platform users generally depict all of those users who, throughout the corporation, use Markman for the purposes that do not directly involve any selling of cranes or crane parts, in example the research and development department. [34]

3.3 Other sales systems

Although, Markman is the largest crane sales configurator that is currently in use at the company, it is not the only one. Heavy lifting business area uses a sales configurator named EOTMan to configure cranes. EOTMan was developed between the years 2003-2005 at the research and development department of Konecranes, in cooperation with NSD Consulting Oy, and it is exclusively used to configure heavy cranes. EOTMan provides basic functionality for an offer designer to configure a crane and produce the offer information. Yet, EOTMan is more of an engineering tool that is intended to assist the offer designer in the offer creation instead of being a true sales configurator. It is considered rather difficult to master among the users and it does not provide pricing information for the offers. It also offers a smaller selection of components than Markman and currently most of the cranes that can be configured using EOTMan can

also be configured using Markman, which has led to the current situation where EOTMan is slowly being deprecated in favour of Markman. [15, 35]

Along with Markman and EOTMan, Konecranes uses yet another sales configurator that mainly focuses on chain hoists and smaller cranes, called Chain system. The Chain system is a true sales configurator in terms that it includes pricing information and sales oriented user interface. Chain system is based on a Summit sales configurator platform, which is a customizable sale configurator platform designed by Wapice Oy.

The company also utilises a number of other sales related software tools that are not sales configurators, but assist in order management and function together with the sales configurators and the rest of the applications in the EAI framework. These systems are commonly called Pro-systems and they are server side applications, which are used to monitor sales processes and their flow. Such a system is ProCom, which is a tool for the sales managers who use it to track down the offers and orders. Salesmen synchronize regularly their offer and order databases from Markman to the system, and ProCom contains advanced search and reporting functionalities for their monitoring. Sales managers use the tool to supervise the sales department, and the system contains history knowledge and statistics of all the offered and ordered products. The collected information is usually harnessed to internal reporting and the harmonization purposes of sales data, but sometimes it is also used in the product design to figure out what kind of cranes are most often requested. [15]

ProFlow, on the other hand is a system that is developed for the following and monitoring of crane manufacturing projects. Project managers mainly use the system as a tool to monitor the overall progress of a crane-manufacturing project. The system provides information about the delivery and manufacturing schedules of components, and possible delays in the production. The ERM system feeds the information to ProFlow on hourly basis, and when a product is offered using Markman, the system automatically creates a frame number for the crane, which is used to link information between the ERM system and ProFlow. ProFlow system reports possible delays in the project elements using different colour codes, and it is used to provide information depicting if the project stays in the schedule. [15]

ProBal is a system that is designed for balancing loads in production sites and it receives its information from the ProCom system. The system is used to control and estimate load balance of different manufacturing sites, and to level the workload between the factories. ProBal system is able to foretell the volume of orders 16 weeks ahead and it can be used to estimate the need of different components and resources in near future, so that the purchases and manufacturing can be planned ahead. [15]

3.4 Sales processes

Sales processes provide systematic methods for performing a product or a service sale. Standardized methods in the sales events helps to reduce the risks of misunderstanding the customers' needs and generally makes the sales event more predictable. Konecranes has defined a number of sales processes for different sorts of sales events, and the sales configurator plays an important part in all of the processes, which is why the future sales configurator must be able to fit to them as well. The products that can be configured automatically with sales configurator follow the sales processes named SP11 and SP12. Another sales process, named SP13, is followed in those situations when a crane cannot be fully configured and platform-level offer specific design is needed. Even though the configurator designed in the future cannot most likely cover the entire product scale of the corporation, one of the preliminary requirements still is that the handling of the special orders should be done in a controlled manner using the application. This would actually expand the concept of a sales configurator into a complete sales system that is able to controllably handle the information exchange for all the different types of orders. [34, 36]

3.4.1 Sales processes SP11 and SP12

Sales processes SP11 and SP12 are presented using UML sequence diagrams in Appendix 4. The process flow in a high abstraction level between the processes is similar because the main difference between the processes is that the SP11 cranes have only one main girder and the SP12 cranes come with two main girders. Both of these types of orders have no specialities and they can be configured using a sales configurator. [34]

The process begins when a customer gets in to a contact with a frontline salesman asking for an offer of a new crane. The salesman analyses customer's demands and configures an offer for such a crane. The configuration also involves the creation of offer drawings that contain all the necessary information needed to build a fully functional crane. Once the offer is created and the salesman has presented it to the customer, the offer information is synchronized to the ProCom system so that the salesmen and management can follow which one of the offers eventually turned into an order. Should the customer accept the offer, an offer confirmation is generated using the configurator, and the information about the accepted offer is synchronized to the offer manager system which also forwards it to the project follow-up system. Once the new order is created, the salesman one more time uses the sales configurator to create the final acceptance drawings and sends all the order information to an order-handling mailbox of the competence center. [34]

At this point the order is ready and it is sent to the production. Competence center picks it up from the order handling mailbox, opens it using the configurator, and splits the order information into component orders and transfers them to the ERM system that also updates the project follow-up system, ProFlow. The competence center also receives confirmation letters from the component orders and the crane order from the ERM system that it forwards to the salesmen who eventually send them to the end-customer. In the meanwhile, the competence center sends out the manufacturing drawings to the crane factories, and eventually the entire crane order is sent to the ERM system for purchase orders and thus the manufacturing begins. [34]

3.4.2 Sales process SP13

The sales process SP13 depicts selling procedures for more complex cranes that cannot be configured and calculated automatically, and which most often require a lengthy requirements gathering process. Appendix 5 contains a diagram that presents the sales process flow starting from an initial contact from a customer to a point when an offer is created and accepted. Appendix 6, on the other hand, presents the sequence flow that takes place after previous sequence to the point when the final product is handed out to the customer. Sales process SP13 is a long process that still requires a lot of manual work when different departments and user entities are communicating with each other; if the new sales system would support a controlled handling of SP13 sequence it could speed up and ease up the entire process. [34]

The process begins when a customer asks for an offer of a special crane from a salesman. Since the offer for a special crane is complicated, the salesman consults the project management for decision making whether to continue building even an offer. In some cases the customer requirements may be so difficult to achieve that it is not financially feasible to start even making an offer. If the requirements turn out to be something that is worth implementing, the salesman contacts the competence center which steers offer design department to calculate an offer for salesman. At this time the offer design department also uses configurator and the DAS server to generate template drawings for a special crane upon which the specialities are implemented. [34]

When the offer is calculated, the salesman presents it to the customer and synchronizes the offer information to the offer management system. Usually with more complex cranes the customer needs more time to evaluate the offer, and finally when the customer decides to go on with the ordering, the decision gets updated to the offer management system, and competence center compares the offer to the newly born order. Acceptance phase begins at this point and it consists of sending offer acceptance drawings between the competence center, the crane salesman and the end-customer. The

end-customer may require slight changes to the drawings, which competence center needs to re-evaluate and respond by sending back updated acceptance drawings for the offer. This phase is repeated as long as it is necessary to gather all the requirements from the customer. Finally, as all the requirements are collected, the final acceptance drawings are sent in a form of an order to collective order handling mailbox, and the information gets synchronized to the offer management and project follow-up systems. [34]

Now the order is ready and it has entered the order handling at the competence center and 'offer to delivery' part of the process, depicted in Appendix 6, begins. The competence center verifies the order one more time and holds a start-up meeting for the project together with the salesman. Sometimes this start-up meeting may even be held before entering the offer acceptance phase if the order appears to be very complicated. ProFlow fragment begins next when a new crane project is activated to project follow-up system. The competence center verifies that the customer has completed the down payment for the order and contacts the design department who commence designing the special features. The design department returns the final acceptance drawings for the competence center and the frontline salesman presents them to the customer for endorsement. After the customer has signed the final acceptance drawings, the competence center creates a component order to the ERM system using the sales configurator. For the fact that the special cranes cannot be fully configured, the competence center finalises the component order manually. [34]

After the design department is finished with designing the special features needed for the crane, the manufacturing drawings are sent back to the competence center. The competence center creates a complete crane order to ERM system together with the manufacturing drawings, from where the crane factories that manufacture the actual crane parts can access them, and the crane enters the manufacturing. The competence center also sends an order confirmation to the salesman who confirms the crane erection schedule together with the customer. Finally when the crane parts are finished, the crane factory delivers the crane to the customer, erects it and hands it out. [34]

4 REQUIREMENTS ANALYSIS

Sales configurator has a large business impact for the company, which is why changing an old system into a new one requires careful planning and preliminary study. A separate business study was conducted to evaluate the needs for new a sales system prior technology platform evaluation, and it consisted of a survey among different users and user groups to determine the desired features for new system. The results of business study, which lead to key feature requirements, are presented in this chapter together with an outline for the requirements specification.

The requirements specification outline follows loosely IEEE Standard 830, which depicts recommended practices for the software requirements specifications. The standard describes the content and qualities of a good software requirements specification and it is aimed to assist in specifying qualities of software that is either developed or purchased. [37]

In spite of the careful planning, a sales configurator project does involve several risk factors that should be acknowledged before beginning the project specification. A separate risk evaluation meeting was held to determine the largest risks for the project and the outcome of the meeting, together with several smaller risks, are also presented in this chapter to complete the requirements specification structure.

4.1 Business study conducted for requirements gathering

The business study focused on the needs that the future sales system and sales processes should be able to answer. The study incorporated theoretical and empiric research about the different needs that the most feasible sales configurator should offer, and was a topic of another thesis. The empirical part of the study consisted of an interview study where the different user groups were questioned about satisfying and unsatisfying features of current sales systems, as well as the expectations towards the future system. The questions also inquired the opinions about the offline functionality and any other development ideas, such as a mobile functionality. The survey was sent to several different users who represented the different user groups of the system.

The resulting answers were coercively collected to three separate groups reflecting different user groups as follows: sales management and salesmen, sales support team, and platform supply chain. In many parts, the resulting answers to the inquiry were conflicting with each other and some of the requested features were even readily

available in the current sales system, which was interpreted as being a problem in the usability and training of current system. The survey included all the different sales systems used in the corporation but only the answers concerning Markman are presented here briefly.

When the users were asked about good features of Markman, the salesmen and sales management listed the DAS drawings and the possibility to make direct EDI orders as most satisfying features. Also the fact that the configuration is exact and allows the user to specify a great number of parameters was seen as a good thing together with the connectivity that Markman has to the other systems, most notably Pro-systems. Many salesmen already pointed out at this phase of the survey that the offline functionality is a good feature, even though there was a separate question assessing the need of it. Sales support user group enlisted mostly the same features as the salesmen but also added as a good feature that a single salesman is able to handle entire sales process and the crane component packages, pricing, technical information and offer drawings within minutes. [15]

The second question dealt with unsatisfying features of Markman. The salesmen and sales management listed the lack of multibranding support and user-friendliness as the biggest flaws. The configurator was seen as too engineering-driven and the offer letters together with the rest of the documentation were thought to be missing professional and sales-oriented look. The engineering-driven approach is also visible in the user interface, which has grown in complexity when the system has been adapted to incorporate larger amount of products than it was originally designed for. Other remarks about the problems in the user interface pointed out that it is a compromise between the alpha and beta business strategies and lacks the description about selectable options. Updating the system was also considered as being cumbersome, since keeping the system updated requires manual downloading and installation of update packages. Sales support group added that updating the product definitions is a tedious task and that the information exchange between the PDM system and the sales configurator is not utilised to the extent that it should be. Both of the groups criticised the lack of centralised customer resource management features, and platform supply chain group added that the manufacturing is not able to support all the configurable features, because of lack of information exchange. [15]

The third question was about the directions that the sales configurator development should take in the future. The salesmen and sales management expressed as their opinion that all the different products ought to be available in one sales system and that alpha and beta business strategies should be separated from each other distinctively. The most common request by the salesmen was an easy-to-use Windows-based application

that includes automatic update functionality and possibly personalized user interface for users with different skill levels. The salesmen also requested a fast web-version in addition to the standalone version, and a possibility to control the entire sales process by using just one tool. [15]

In contrast, the sales support group only saw need for a web-based version that could be kept up to date more easily. It is, however, notable that the group of people that only saw need for a web-based version do not use the configurator in their daily work. The sales support group also expressed their wishes for one uniform sales system, instead of having many separate tools for different cranes. The platform group saw a major development scenario in the way that the new product information is updated to the configurator. Currently the information travels backwards in a way that the new product information is first stored in the sales configurator and the product drawings are implemented into the PDM system afterwards. The ideal situation would be such that the sales configurator would automatically update the sales BOM information after the engineering BOM information from the database. [15]

The business study and survey included many other relevant questions that are not covered in this text. Many recurring wishes contained a missing CRM support and a possibility to configure a bigger number of different products with the sales configurator. The users also expressed their willingness to have a single sales configurator for all kinds of products instead of dividing the configurator project to smaller sub-projects, which might have been an option, for example, in component configuration. The majority of respondents, when specifically asked, also supported one of the most controversial features, offline functionality. On the other hand, mobile phone or other handheld solutions were not seen important. The users emphasized also the need for better training, which may have reflected the complexity of the current user interface. [15]

The survey was conducted to each different sales configurator, the company uses at the time, and Markman received generally the most positive feedback out of all of the configurators. The questionnaire also brought out doubts whether the new system will be able to live up to the old system and meet all the new expectations. One of the worries was that a web version might perform in slower and clumsier manner than a native windows application that the previous application is. Also an important factor in the future version was seen that all the existing functionality of the old system should be available in one form or another in the new system. [15]

The preliminary business study concluded with the thoughts that, in the future, there should be only one sales system for all the different products, which would not only

ease up the maintenance but also help to synchronize all the information that is currently scattered among separate systems. Attention should be paid to turn the new configurator into more customer and sales oriented direction whilst not forgetting the demands that the different brands have for the look and feel of the application. At the end of the day, having a single sales system with a centralized database would point towards the unified company image that the corporate management pursues. The new sales system should also be able to support an integration to the CRM system that is planned to be coming in the near future, and it should be designed so that the crane related technology knowledge that exists in the company could be harnessed in to the sales configurator software development process. [15]

4.2 Identified risk factors

A risk evaluation meeting was held to determine the largest risks for the project. The meeting approached the risk factors in a high abstraction level and did not include technically specific details since the actual project and its outlines were yet to be determined at that stage. The risk identification meeting focused on understanding the environmental context of the configurator project and the impacts that acknowledged problems would have to the company and its business operations. By recognizing the risks in advance, the actual development of the project can be organized in such a way that special attention can be allocated to those areas of the project that have the largest financial impact.

The largest recognized risk factor has so tremendously large financial impact for the corporation that it also poses special requirements for the project planning: the sales configurator must not under any circumstances calculate wrong results. Currently the sales configurator handles the calculation of steel structures and the creation of large part of manufacturing information automatically, and if there would ever be a situation when this information would be wrong, the results would be devastating. Markman currently calculates the strengths of the steel structures and ensures that the resulting crane that the salesman calculates can always be built safely. If there would be an error in some parts of the calculation, a salesman could not possibly manually identify the wrong results, as the calculation of strengths in the steel structures of a crane is a task that would require several days even from a trained mechanical engineer.

Because the old configurator has done all the strength calculations and verifications that are required to build a crane, all the rest of the enterprise systems provide no extra validation for the input data, as all of the orders coming from sales configurator are assumed to be correct. So far, the current system has never failed in the calculations. In

the worst-case scenario, a software bug could affect the strength calculations in such a way that the entire crane would collapse. In addition to the damage that 80 tons of collapsing steel would do, it would also mean that all the crane manufacturing should be halted instantly and the delivered cranes should be re-examined. This scenario would have such a devastating influence in both financial and humane point of view that the precautions preventing this kind of scenario ever happening have number one priority in the design phase.

The risk that the order information is altered or changed during the order-transferring phase from the sales configurator to production is smaller. Even if the company would become a target of a malicious attack, where a hostile attacker would be able to alter the information between the configurator, the order handling and the order receiving competence center, the results would not be so dramatic, since all the offers are re-calculated at the end point before the drawings are sent to the production.

Another risk that has a high business impact is that the work of the salesmen should not stop at any point. Even though the requirement for the new system to work properly is rather evident, the risk that the new system does not contain all the features and configurable products exists. This risk is, most of all, a project coordination risk that should be taken into account when the new system is deployed, although the product fulfilment verification and validation is something that needs to be taken into account even throughout the design phase. The salesman should not lose a single bid because their system is unable to configure certain crane configuration.

Validation and verification are commonly used terms in the software industry that try to answer the two different questions "*Are we building the product right?*" and "*Are we building the right product?*" The first question issues the technical details about the completeness and functionality of the application and the second question issues whether the application that is designed actually suits the purpose it was intended to. Although, the simple requirements for validation and verification may feel obvious, the nature of the future sales system is so complex that transferring all the requirements to a form that a software contractor can fully them understand, poses a challenge. This is an issue that can only be solved by listening to the end-users while developing the configurator to ensure that the new version does not perform in inferior way to its predecessor. Even though the requirements are studied very carefully, a popular catchphrase in software industry states that the company, which underestimates the size of the project worst, usually wins the software contract.

From the business point of view, one risk concerns the pricing information and its safe storage. Currently the different brands have their different and separate installation

packages in order to prevent information leaks between the brands, but a larger issue concerns leakage of business information outside the corporation. As the sales configurator has more than 1600 users worldwide, many of them operating outside of the corporation, the risk that a hostile party attempts to reverse-engineer pricing information exists. The problem does not only concern the pricing databases and the technical calculation but the entire programming logic behind the pricing and crane optimization routines, since if a hostile competitor would be able to compute prices of Konecranes' crane offers, it would be able to take over all of the crane bidding situations and eventually cause drastic losses of sales.

Even though industrial espionage and reverse engineering may sound like a far-fetched ideas, according to a survey carried out by Finnish Security Police (Suojelupoliisi, SUPO) about intelligence activity against the companies in 2007, approximately 10 per cent of the companies that responded to the survey informed that they had been targets of illegal information gathering within the last two years. Although, the main focus in the espionage is on the new and emerging fields of science, interest is also shown towards the cooperation between research and business life, which might affect Konecranes and sales configurator as well. [38]

This risk is increased because all of the updates and distribution of the configurator software must be done in public information networks, for many of the users are working outside the corporation. Currently, running the update requires an exchange of authentication tokens, known as installation codes, to prevent unauthorized downloading. This authentication token method, however, was found to be cumbersome to use in the business study, and the general atmosphere requires looking for more automated alternatives which may be difficult to achieve without compromising the information security.

In addition to the business related risks, the realisation of the project involves several software development related risks. Software development risk signifies a risk that prevents a software project to finish as planned, basically meaning that the new software is unable to meet the development goals, the costs are higher than planned, or that the project cannot be finished in schedule. The software development risks are already old research topics and a great amount of literature exists for their assessment and proper management.

According to the literature, software risk management is considered to be on-going activity that spans through the development project starting from the very beginning, and contains identification and classification of the risks, together with plans for their avoidance and mitigation. The risk management is supposed to be a continuous and

periodic review of the different risk factors, where the changes in the probabilities and impacts of recognized risk factors are assessed as the project matures. All of the risks cannot be reduced or completely avoided, thus a proper risk management contains also backup plans for those inevitable moments when problems occur to prevent project organisation to fall into crisis atmosphere. [29]

In the early nineties, Barry W. Boehm identified ten most important software risk factors in his science articles that up to this day have made it to the most of the software science books. According to Boehm, most of the post-mortems of software project disasters have indicated that the problems could have been avoided or strongly reduced if there would have been an explicit early concern with identifying and resolving high-risk elements. Frequently, the projects that ended up badly were started with overly optimistic enthusiasm during the early phases of project that caused them to miss some clear signals of high-risk issues that provided to be their downfall later. Boehm answers the start-up speed frenzy by employing a set of formal methods to follow in order to find those factors that may result to problems and realisation of risks. Boehm's methods include identification of the risks, calculating probabilities for their occurrence, and prioritization of the risks accordingly. To ease up the identification of different risks, Boehm collected a top ten list for most common mistakes using surveys to be used as a checklist by project management. The most common risk items in software industry with brief explanations are listed below:

- *Personnel shortfalls*, which means a lack of qualified personnel in the project team. Also, items like insufficient expertise and unrealistic expectations of personnel's abilities can be categorised as a personnel shortfall. Personnel shortfall may also occur when personnel's traits are not properly mastered putting the individual workload between different team members out of balance, or when key persons leave the project.
- *Unrealistic schedules and budgets and underestimating problem complexity* mainly deal with misunderstanding the complete problem domain and overestimating the abilities of personnel. This problem is likely to happen in conjunction with the other problems such as transferring exact needs of external software components to a subcontractor or straining the limits of computer science.
- *Developing the wrong software functions* is a problem that can only be solved by listening to the end-users and analysing the needs of the organization. Other useful methods for dealing with the problem include building early user's manuals and prototyping the product.
- *Developing the wrong user interface* is an obvious problem but still very rarely pursued enough. In the end, the user interface is the mean how users access the

system and if overlooked even the best of the applications is crippled. User interface flaws can be detected by prototyping, task analysis and general end-user participation to the project.

- *Gold plating* means adding unnecessary bells and whistles to the project with the purpose of making it look good in the eyes of the end-user. Not only does the additional and unwanted features cost extra, but they also add the complexity and the number of possible faults in the software. Performing cost-benefit analysis and requirements scrubbing for the different features of the software can be used to identify the gold-plated functionality.
- *Continuing stream of requirements* means that the requirements are constantly changing throughout the project, and altering the application according to them is often described as hard as hitting a moving target. Constantly changing key requirements are not exclusively results of an incomplete requirements gathering but may also indicate too low threshold in the requirements management.
- *Shortfalls in externally performed tasks* deal with subcontracting and problems relating to it. The risks involve poor quality or unpredictable accomplishment of the tasks that are performed outside the organization. Transferring the exact needs to a contractor is a part of the problem that can be solved by reference checking and pre-award audits.
- *Shortfalls in externally furnished components* also deals with the problems in subcontracting and relates to the problems in components that are delivered externally. These issues can be reduced by benchmarking, inspections, reference-checking and compatibility analysis.
- *Real-time performance shortfalls* occur when the system is unable to function in a speed that is required for its appropriate usage. Sometimes the new software can perform slower than the previous version due to the new features, affecting the overall user experience. Simulation and prototyping can be used to measure the actual usage capabilities, and fine-tuning can be made to speed up the application.
- *Straining computer science capabilities* means inability to implement the system because of lacking technical solutions and computing power. Some problems are such by nature that they simply cannot be resolved using the information that is available, and stretching the system to meet the demands may result in clumsy implementation. These risks can be recognized with technical and cost-benefit analyses.

[39, 40]

Boehm notes that most of the critical risk items in the checklist have something to do with shortfalls in domain understanding and in properly scoping the job to be done. However, another science article notes that none of the risk components are solely

technical; instead, technological aspects are embedded into the software risk components and have always human actors and understanding in the background. [39, 40]

4.3 Selected key requirements

After a careful judgement of business study and risk evaluation, the configurator steering group selected nine of the most anticipated features as the primary requirements for the new configurator. The primary requirements are not intended to cover all the demands exhaustively; instead they represent the most important qualities that are required from the new system in addition to the basic functionality that the old configurator offers. The following key requirements are not presented in any particular order:

1. Possibility to have at least two or three different user interfaces to serve users with different skill levels. The basic user should find the system easy to use and the user interface should be kept simple, whereas more advanced users could be given more available selections and the user interface can be more demanding. The company has also plans to start arranging education for salesmen to harmonize their knowledge of sold products. One idea in the education is that once salesmen attend the courses and learn more, they will become eligible to sell more complicated cranes. The user interfaces could be tied together with the training levels.

2. Possibility to have both, offline and web version of the software, and if the two versions would look similar, it would considered as an advantage. The strong demand for offline use that came from the end-users cannot be overlooked and the offline functionality should be implemented. However, with careful planning, a web version could be built using almost the same code base that the locally installed version uses. Web version would be ideal to serve more casual users to limit the amount of desktop installations of the software. It could also be updated and maintained at a faster pace than a desktop installation.

3. Integrations to other systems, such as the ERM, PDM, Pro-systems and global CRM. Even though the global CRM system is not implemented yet, it will most probably be one day. Thus, it is important to build the new sales configurator in so that it can be integrated to the other systems with a minimum effort using standard interfaces for information exchange.

4. The system should be able to read product information from the PDM system automatically. Current situation where the different measures from engineering product structure are copied manually to the sales configurator database is unbearable. The number of configured products is only increasing in the future, as the responsibility of the sales configurator is extended, and keeping the sales product model up-to-date is only getting more difficult. Since the PDM system already contains the product structure information, it should be read automatically from there. Automatic reading of sales BOM information from the PDM system would also answer to the problem where the sales system supports products that are not yet even completely designed by the engineering department.

5. Multibranding ought to be supported better than it currently is. The configurator needs to serve different brands according to the business model of the company, and the different brands require having their own look and feel in the application. Currently not only do the user interfaces look almost alike between the different brands, but also the printouts and documents that the system produces look similar to one another, the only difference being a change in company logo at the top corner of a document, which does not qualify as multibranding. The different brands would like to get more freedom to affect the outlook of the documents and the application.

6. Easiness of maintenance. The configurator should be easily maintained in terms of the product structure as well as in terms of keeping the software updated. Software updates should work automatically, although special attention needs to be paid to a secure update mechanism to prevent information leaks.

7. Technology that does not age very fast. The expected lifespan of the new sales configurator system is required to be at least ten years. During this time many new systems will most likely be introduced in the company, and the entire operating environment of the application is likely to change as the technology of workstations evolve. It is important to select the kind of a technology that will be usable after many years, so that no options would be closed because of wrongly selected technology, when the new requirements for the system rise in the future. In a best-case scenario, all the work that is put to the new configurator software today can be utilised in the following generation configurators tomorrow.

8. The web version must be integrated to Konecranes portal. The corporation policy dictates that all the new web applications are embedded to a corporate wide IBM Websphere portal. The main idea is to have all the different web applications in one personalised web space seamlessly integrated to each other, so that the users need not even know that they are using separate applications. The requirement does not state that

the web configurator has to explicitly run on the same server as web portal but it should be integrated to it.

9. Reusing the old software code to the extent that it is possible. This requirement follows from identified risk factor number one, that the configurator must never calculate wrong results. The intent behind recycling software code is that the crane specific calculation modules are kept untouched and integrated to the new configurator. This serves as a risk-reducing factor and helps to mitigate the largest identified risk: the miscalculation of crane steel structures. Having the new configurator to utilise the old calculation components makes it possible to develop the configurator in smaller steps, and during the development phase the old calculation components can be trusted to always provide the correct results to ease up the testing.

Keeping the calculation components intact also serves as a mean to separate the crane related knowledge from the configurator related knowledge, which needs to be done in any case, as requiring crane knowledge from a software vendor is practically impossible and far from desirable. Most of the crane expertise in the corporation is built inside of the calculation components and many of the components are in fact complete calculation applications that have only been compiled into form of ActiveX components. Rewriting all of them at once would require tremendous effort and could result in unpredictable problems. Now as the calculation modules are kept intact, the development can be focused on the sales configurator, and the calculation modules can be redesigned one-by-one afterwards thus reducing the risk of design problems.

4.4 IEEE STD 830 Software Requirements Specifications

IEEE standard 830 depicts the recommended practices for creation of Software Requirements Specifications (SRS). It describes the content and qualities of a good SRS and presents sample outlines for their creation. The standard is aimed for specifying the requirements of software to be developed, but it can also be applied to assist in the selection of in-house and commercial software products. It does not specify any method or a tool for preparing a specification; instead the standard focuses on outlining the different aspects that needs to be taken into consideration when the software requirements specifications are constructed. [37]

SRS is meant to help the software customers to accurately describe what they wish to obtain and the software suppliers to understand exactly what the customer wants, thus a good SRS helps to establish the basis for the agreement between the customers and the suppliers on what the software product is to do. It also helps to reduce the development

effort as a careful review of the requirements can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when the problems are easier and cheaper to correct. SRS also provides baseline for the validation and verification as the created software components can be compared against the original requirements specification, which also serves as a basis for enhancement of the developed software because it discusses the product but not the project that developed it. [37]

The standard suggests that a good SRS should address at least following topics:

- a) *Functionality*. What is the software supposed to do?
- b) *External interfaces*. How does the software interact with people, the system's hardware, other hardware, and other software?
- c) *Performance*. What is the speed, availability, response time, recovery time of various software functions, etc.
- d) *Attributes*. What are the portability, correctness, maintainability, security, etc. considerations
- e) *Design constraints imposed on an implementation*. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environments, etc?

[37]

In addition to addressing the listed topics, the standard poses requirements for how the topics in the SRS should be covered. A good SRS needs to be unambiguous in a sense that every requirement must have one and only one possible interpretation and the terms that might have multiple meanings ought to be included in a glossary where its purpose is made specific. Ambiguous statements follow from the fact that the SRS documents are written in natural languages that are inherently ambiguous, thus it is preferable for a SRS to include diagrams and standardized presentation formats for clarification purposes. The SRS should also be complete so that it includes all the significant requirements of the software together with the definition of the responses to all realizable classes of input data in all realizable classes of situations. [37]

Other requirements for a good SRS include that all the statements in the document are verifiable. A requirement is verifiable only if there exists a finite method to evaluate whether the software meets the requirement. Nonverifiable requirements include statements such as "works well", "good human interface" or "shall usually happen". These are statements that cannot be verified because they do not use concrete terms or measurable quantities. In addition, all of the statements should be also traceable, which means that the origin of each requirement is clear, and internally consistent. [37]

As the main idea behind a SRS is to find a uniform understanding between the software purchaser and the developer, and to ensure that the requirements are transferred to the developer in most unambiguous way, a joint preparation is needed when the specification is created. The software development process should begin with supplier and customer agreement on what the completed software must do, and the SRS is the tool for the mutual understanding. It is important to acknowledge that neither the customer nor the supplier is qualified to write a good SRS alone. [37]

4.5 Brief supplementary SRS outline for sales system project

Even though this thesis answers to many topics that should be covered in a good SRS, many parts of the project are still explained in too vague a manner for this thesis to be considered as a valid SRS. Nevertheless, one of the purposes of this thesis is that it could be used as a template when the SRS is created jointly with the selected software vendor, because many parts of this thesis deal with the same basic issue that a SRS does: transferring customer needs to a third party software developer. The purpose of this subchapter is to expand and specify requirements and limitations that were introduced earlier to the extent that they could be used as a basic outline when IEEE Standard 830 compliant requirement is written with the selected software vendor.

This subchapter itself should not be considered as a valid SRS for a number of reasons. Because many parts of the key requirements describing the problem domain and the definitions for different environmental conditions are described elsewhere in this thesis, they are not exhaustively covered in this subchapter thus violating the rules of completeness. Instead this subchapter discusses those topics that ought to be covered in SRS and are omitted in earlier chapters. The subchapter should be treated merely as an outline since creating a competent SRS before even a software vendor is selected would be impossible and clearly out of the scope of this work.

4.5.1 Description of desired functionality

The new sales configurator should include a separate web application and a client application much in the same way that email applications offering separate clients and web interfaces are implemented. The client version should be able to configure cranes even without network a connection, although the full functionality needs not to be attained in those conditions. The web application must be integrated to the Konecranes business portal and the client application must work on a Windows desktop environment. The client application implementation must follow the recommended practices for creating MS Windows applications so that the configurator functions

correctly also in the future operating systems used in the company. The user interface needs to be customisable according to the needs of different brands and differently skilled users. Different brands need to have an ability to customise the visual appearance of the user interface so that it looks different from other brands. The same requirement applies to all of the documentation, such as offer letters, that the configurator produces. Differently skilled users must have a possibility to have distinct user interfaces that match their level of competence.

DAS system is intended to stay intact and will not be redesigned for this project. Should the new configurator platform bring new additional constraints to the information representation format that is sent to DAS server, the DAS server software can be altered to read the input data from some different file format. Out of all Pro- systems, the configurator only communicates with ProCom offer management system and uses a HTTP (Hypertext Transfer Protocol) connection for information exchange.

The system should be designed in such a way that a user cannot make wrong or inconsistent selections, as there are no other systems in the order to manufacturing chain that ensures the validity of the order. In terms of performance, the web version must be able to handle at least 50 simultaneous users performing crane calculation, and in addition to calculation, another 1350 users must be able to operate the system doing less demanding tasks. The system must be scalable for larger amount of concurrent users in the future by adding new server hardware. The actual hardware specifics are not defined yet in this preliminary technology study, for the development time is most likely to take years during which time the computer hardware technology presumably changes. However, the requirements should be met using an averagely priced business server, most likely built as a part of a larger server rack in a corporate data center.

The software code should be portable to the extent that is possible. The ActiveX components that are used in the calculation of steel and crane structures limit the portability of the application, but in the future if the calculation modules are rewritten using a more standardized and neutral technology, it would be considered as an advantage if the system would be able to operate platform independently. Maintenance of the crane calculation components and crane related knowledge should still be left to Konecranes corporation, and the system must be designed in such a way that the new and existing product families can easily be maintained by crane specialists that may not have knowledge of computer programming. The IT-specialists of Konecranes are able to do basic maintenance operations, such as rebooting the system and performing other failure recovery operations. The system should also have an open external interface so that the corporate software developers are able to implement other supplementary applications that communicate with sales configurator.

The user groups of the application will be similar to the user groups of the old system, with the exception that the user privileges should be designed so that they can be restricted from certain users or user groups dynamically. Different users must also have an access to each other's information, although a representative of another brand or licensor must not have an access to the customer, sales and pricing information of other brand. It is mandatory that the users can be divided into groups that are able to see the information of different users inside of their own group but cannot have access to the information of another group.

All of the constantly running systems, such as web applications or back-end services of the application must be able to recover from most common failures by themselves. This demand for failure recovery depicts situations, such as automatic recovery of lost database connections or other problems that may arise from temporary network problems or power outages. In an ideal situation, the system should recover without manual intervention, once the underlying problem gets repaired.

4.5.2 External interfaces and information exchange

Since the sales configurator includes a web-version and a standalone version, both types of user interfaces are needed. The user interfaces ought to be designed in such a way that building a configuration variant of a simple basic crane should not take longer than a minute. The user interface must include such default values for different user inputs that whenever user decides to finalise the offer, the crane can be safely constructed. The user interfaces should also contain templates that set the initial values of input fields for different types of well-known cranes to speed up the configuration selection process. These templates can be, for example, most common usage scenarios of cranes that have similar characteristics. These templates must be modifiable and designed so that the system administrators are able to build more of them and edit the existing ones.

The sales configurator needs to have connections to external information systems, software interfaces. The old configurator connects to DAS server via HTTP file upload mechanism that is built into the DAS system, and sends all the configured parameters in a form of a single ZIP file which contains an Access database that includes all the variables that are needed to create a 3D drawing out the of the configured crane variant. This information transferring mechanism can be changed in a new version of the sales configurator if necessary, but basically the information that is required by DAS still remains the same. ProCom information exchange works in a similar way than DAS system as a ZIP file containing a database of needed information is sent to the receiving server application.

Most notable external software interface is the EDI transferring interface. All the orders coming from the sales configurator are transferred to ERM system using EDI protocol. The information that the sales configurator needs to generate in order to define a unique crane variant consists of two different parts: DAS drawings, which are needed in crane factories to build the steel structures, and technical statements that define different parameters needed for the rest of the crane specifics. Currently the sales configurator generates a number of configurator specific variables that are sent to the ERM system using EDI protocol, which are then read and converted into production specific technical statements using a separate EDI parser tool at the ERM side of the system. Technical statements depict all the different requirements, such as hoisting motor and gearing information, that are needed before a crane can be built, and generally define unambiguously a single crane variant. The DAS drawings, on the other hand, are needed for the assembly of the crane and in the construction of steel structures.

The new system is allowed to change the EDI parser and the entire protocol that is used to transfer the information to ERM system, yet the similar functionality is required to exist also in the future. In an ideal situation, the order information that travels to the ERM system would also contain DAS drawings in the same bundle, and that during the EDI transferring phase the 3D drawings would be generated automatically. EDI is one of the oldest standards for depicting structural information, and as the world has seen a rise of new XML based information exchange methods during the recent decade, it would be recommended to change the EDI protocol into a newer one.

The old sales configurator is also able to store all the configuration information in a ZIP file that contains both, technical variables in a form of a database, and DAS drawings. However, the ZIP files are not opened in any other system except the sales configurator and every time when a crane order is transferred to production, the ZIP order files generated by the old configurator are manually opened elsewhere using the configurator and sent to the production by utilising EDI order mechanism. Therefore, the ZIP file is only used to move information conveniently between different configurator users. In the future sales configurator, the existence of ZIP file is not mandatory, as long as the same functionality, information exchange between the users, can be achieved.

As a new addition to existing information exchange comes the reading of sales BOM information from the PDM system. The PDM system offers an external XML based interface for information exchange called PortalAPI. However, the sales BOM information could also be read directly from the database of the PDM system, which would also help to reduce the overhead and system load that PortalAPI causes. The information that needs to be read for the sales BOM is not readily available for the time being, and it needs to be specified together with the selected software provider before

the new configurator is implemented. Also, the information exchange that takes place between the sales systems and the upcoming CRM system cannot be specified in advance but should be taken into consideration for future implementation.

4.5.3 Constraints and limitations

The configurator implementation comes with several constraints and limitations regarding the usage environment. The workstations that are running standalone client applications are using Microsoft Windows operating system with limited user privileges. As an initial assumption, the desktop version of sales configurator should be a native Windows application that is designed using 'recommended practices' guidelines for the development of application. Following the guidelines ensures the compatibility also in the future versions when the operating system gets updated, and that the installation and maintenance of the application can be done using the most common methods. [41]

The service desk and workstation support in the corporation is handled by an external subcontractor, which imposes additional limitations to application distribution and maintenance. Hewlett & Packard that handles the desktop support for Konecranes requires all the installed software packages to go through their quality examination before they can be remotely installed on user's computers. For the reason that the users are not allowed to install any applications by themselves, also the building of installation packages and the requirements for automatic update system come with extra challenges. All the applications are installed to default locations and during the installation phase, the installer application receives system level privileges. A regular installer application will qualify for the purpose but an installer that utilises MSI (Microsoft Installer) technology is recommended for the task. Other constraints regarding the operating environment involve limitations to communication channels, as the corporate system image used in the desktops restricts a number of communication protocols and ports. [41]

Configuration information may require rapid updates, in example, in pricing tables due to sudden changes in world economy, which is why the application distribution using a preferred centralized method may lead into problems. The original key requirement for keeping the application easily maintained possibly contradicts the limitation of planned centralised application distribution, which is why special attention needs to be paid to this area when implementing configurator. Hewlett & Packard does offer an escape valve to assign special privileges for certain predefined processes, which could be used as a mean to handle automatic updates. The exceptions for the rules, however, are issued and negotiated case-specifically. [41]

Another limitation comes from configuration calculation of different crane structures, which requires exhaustive computing power. Currently, calculating a more complicated crane variant on a modern desktop may take more than a minute whilst utilising all the processor power available. If this same calculation is taken to a server that serves numerous simultaneous users, the responsiveness of the application may fall below usable, which is the reason why the configurator system needs to be easily scalable. The system will have its own dedicated servers located at the corporate data center but the configurator software must be designed so that adding new hardware increases the performance and capability of the system.

Business portal environment comes also with constraints. The implementation of business portal application is a controlled process and comes with predefined guideline documentation. The business portal guidelines do not only cover embedding of an application to the portal but also application architecture and development processes. The guidelines for embedding are inherently restrictive and aimed for the design of smaller web-applications that run within the portal environment, thus the actual portal integration of configurator must consist of compromises that have to be negotiated specifically. The portal guidelines offer also limitations that can be applied to the configurator project directly, and guidelines controlling the visual appearance, for example, can be followed as such. Prior implementation phase, the level of accuracy that the guidelines are followed needs to be thoroughly specified. [42]

5 SUGGESTION FOR SYSTEM STRUCTURE

This chapter appends earlier requirements and functionality definition by introducing a high level architectural example of a configurator software. Transforming the software requirements for a third party developer is task filled with pitfalls, and the purpose of this architectural suggestion is to present a solution that would satisfy all the given requirements and approach the definition using an example solution. The overall approach is kept artificial and it may not represent the final solution as the third party software vendors are expected to introduce their own architecture solutions. This suggestion for system architecture is neither a customer demand nor does it cover any usage of time and resources.

5.1 Basic structure

Having the two separate versions, a web application and a Windows client challenges the application architecture when trying to minimize and recycle the code. Building up two entirely separate versions of the application would become handful to update and keep maintained, thus the architecture suggestion is designed in such a way that a minimal amount of code would be needed to build the two separate versions of the application. On a basic level, the suggested architectural structure follows conventional three-tier architecture in separating three different layers, presentation layer, application functionality and storage of persistent information. Architecture is presented on high abstraction level in Figure 10.

The fundamental idea of the architecture in Figure 10 is that the main application, entitled “Application Engine”, utilises exactly the same source code base in both different versions of the application, and the only thing that is different between the two versions is the user interface. An external database runs on a separate server and its task is to contain permanent information and provide common interface for external operations. In addition to a global database, the application itself contains a local database, which is synchronized to global database when network connection is available. The user interface would offer an adapter interface that encapsulates the functionality of user interface and provides common functionality for building user controls and dispatching the events.

The idea behind the suggested construction is that the same application could be used in the most flexible and reliable way so that it still is able to provide the desired functionality. The application engine is considered as a main application of the

configurator and does not contain any user interfaces itself, but it is designed to use abstract interface without posing any restrictions to the actual implementation. The application engine handles the configuration and calls the calculation modules to do the strength and structure calculations. It also hosts local databases for general product structure and offers an information synchronization service that transfers information between the common database and a local database. All the connections to the external systems are designed to work through one centralized common database, which aims to simplify the structure of application engine.

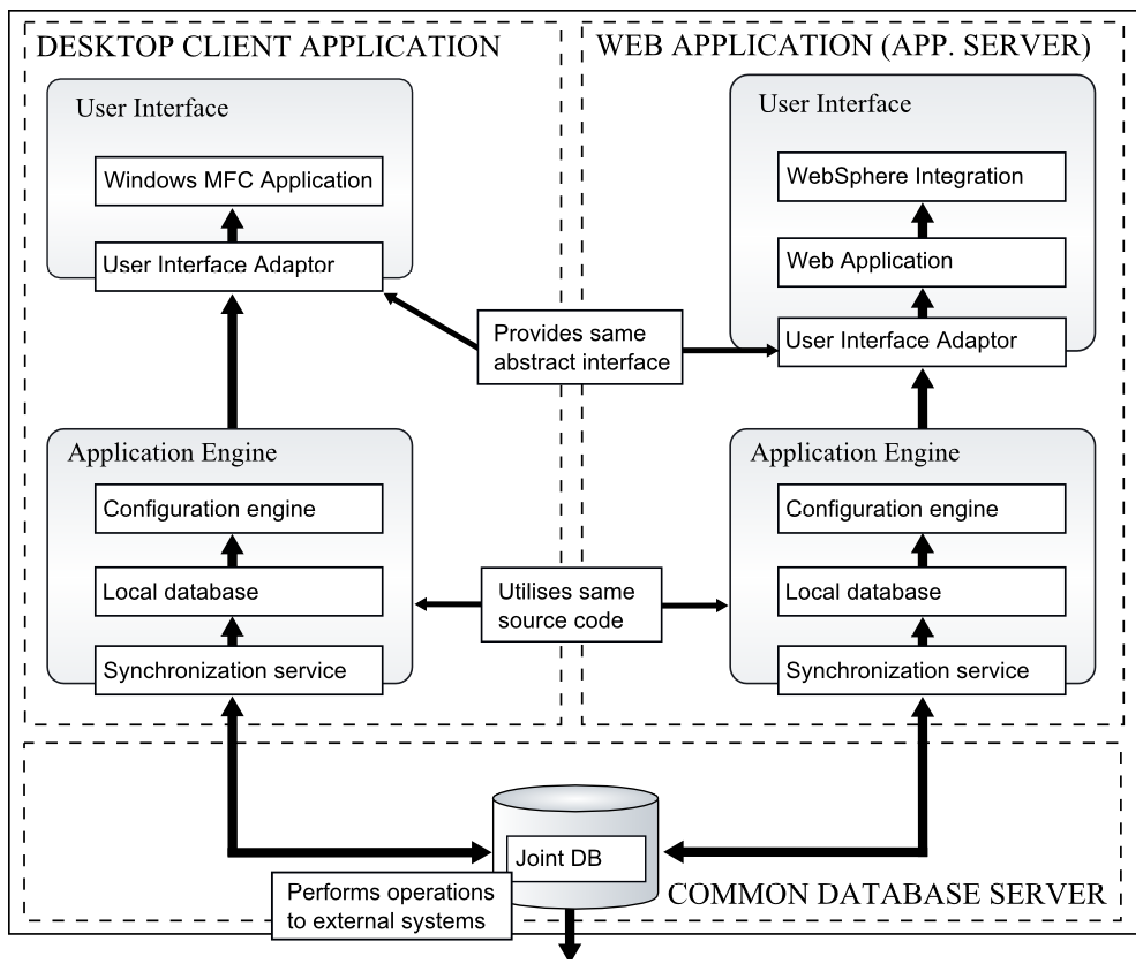


Figure 10. Suggested system architecture in high level of abstraction

Reasoning behind the architecture was to find the minimum amount of software code to do the task for reliability and maintenance purposes. Furthermore, the architecture serves the development phase as the different application layers can be designed independently of each other, and since the application engine only requires a compatible user interface adaptor, new user interfaces can be added to the system as new needs emerge. From the maintenance point of view, all the changes to the main application can be applied first to the web version where the problems are faster to debug and fix,

and since the application engine is compatible with the client version, the altered engine can be updated to the client application.

Other benefits for the development include a possibility to easily build new user interfaces on top of the core application, should any new needs emerge in the future. This could include building such new interfaces as simplified wizard-like versions for the corporate web site for the customers who would like to configure their own cranes, or even mobile applications. Again, as new user interface techniques are likely to appear during the next decades, the external user interface layer would make it possible to port the application to the new technology easily.

As all of the connections to the external systems would be done from within centralized and common database, it would leave an easy door open when new systems, such as upcoming CRM system are created. Centralizing all the external connections to a back-end server would promote the information security so that the external systems do not need to be opened for public access outside the corporation network, instead the back-end server would provide secure authentication and possibly also encrypted and secured communication. Basically the application would function in similar manner as sophisticated e-mail applications do: having a local cached database to enable offline functionality, and several user interfaces built on top of the same application core that are also able to provide required functionality when used from within a web portal.

5.1.1 Presentation tier in detail

Separating two different user interfaces can be made in many different ways but the requirement for web interface poses an extra challenge. Basically the configurator needs to be able to operate as a web server application and since the user interface is what separates the web version from the standalone version, it is also logical to bundle web server operation to the presentation tier. From a software project point of view, the inclusion of two different types of user interfaces could be done in many different ways, such as utilising abstract factory design pattern or even using pre-processor commands if using C++ as shown in Figure 11. An abstract factory pattern is a software design pattern that can be utilised in full object-oriented programming languages. It provides a method for encapsulating similar type of functionality behind one common interface, which basically means, in the case of example, that a same command with same calling parameters can be used to create a push button to the screen, and depending on the project parameters, the push button would either appear on the screen of a standalone application, or on a screen of a web page.

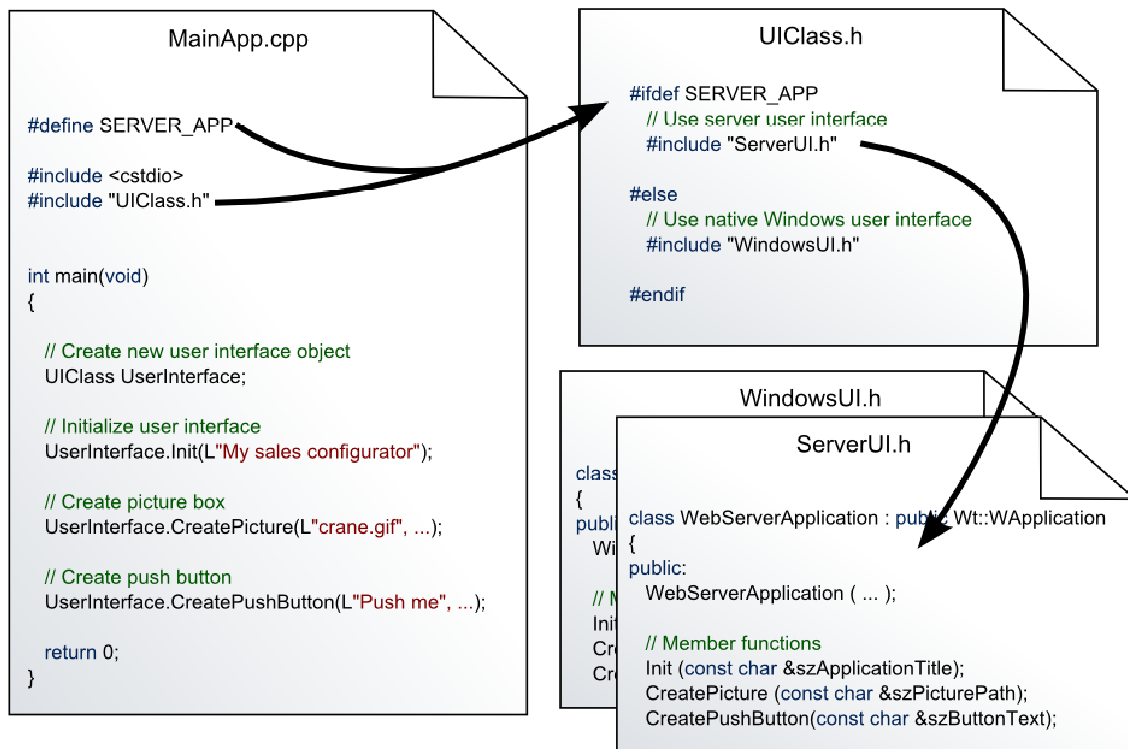


Figure 11. Main application using same code for different user interfaces.

The Figure shows an example where different user interfaces are selected compile time using C++ pre-processor commands. The first line in main application code module defines that server application settings are applied to this compilation. Third line includes an abstract user interface class to the project, which, depending on the application setting, either includes a server interface or a native windows interface to the project. The main application uses exactly the same commands for creating items to the screen, but the true nature of the items vary according to the type of user interface that is used.

For this technology suggestion, a utility library named WebToolkit, shortened Wt, was chosen to provide the desired functionality for server-side user interface. Wt is a C++ library, which provides functionality of an application server and web interface in one package. Instead of being a page-based framework for web application development, like PHP or JSP, the library encapsulates fully interactive and transparent AJAX (Asynchronous Javascript and XML) framework that allows easy creation of interactive web-based user interfaces. Basically, Wt applications need not to be considered as a web applications by developers, instead the library acts as a graphical user interface toolkit which allows users to create buttons and user controls same way that they would be created using a regular graphical user interface development library such as Qt or Windows Forms. The system also hides the compatibility issues between the different web browsers and handles even limited JavaScript support so that the developer does

not need to be aware of web specific technologies that underlie beneath the library, and even incorporates an open source Asio web-server that compiles together with the rest of the application. The library offers also a possibility for tight integration with Apache web server when using Linux environment. [43]

AJAX technology that forms the foundation of the Wt library provides asynchronous information transfer between a web server and the web browser which is used to view the web application. Traditional page-centric web applications need to re-generate the entire web page whenever user interface needs to be updated, which essentially turns web interfaces slow and cumbersome to use. The idea behind AJAX is that the information required to update the screen is divided into smaller pieces and loaded silently in the background in an asynchronous manner. Smaller screen updating packages lead into faster response times, which make the application to resemble more of a native desktop application.

The original intentions behind AJAX technology are not very different from the needs of the sales configurator application. AJAX was designed to close the gap between the traditional desktop applications and web applications by providing interactivity to user interface. It was used first time in Microsoft Exchange e-mail application to make the e-mail client look and behave exactly the same way when using webmail and a separate desktop mail client. The asynchronous XML based data transfer methods quickly became popular in other applications as well, since it is based on open standards and allows cross platform operation. Google Inc, for example, offers full office application suite named Google Docs, designed using AJAX techniques containing spreadsheet, word processor and presentation applications, that all can be used online. [44, 45]

AJAX is not, however, the only way to create rich desktop-like web applications. Several other methods exist to provide similar functionality, such as Adobe Flex or Java Applets; nonetheless AJAX was chosen for this suggested architecture mainly because it is an open W3C (World Wide Web Consortium) standard that does not require any external applications to run the code. AJAX has also been gaining popularity during the recent years and several major software companies, such as Microsoft and Google has invested heavily in its development, thus it is likely to be long-lasting solution for the future. [46, 47, 48]

Out of many existing AJAX frameworks, Wt was primarily chosen for the fact that it contains full built-in web-server functionality which is able to transfer user interaction to underlying application same way that a regular desktop user interface library does. It also offers functionality of common web-based technologies, such as automatic handling of SSL (Secure Socket Layer) and compression of data packet headers.

Basically the system makes it possible to outsource entire web user interface backend so that all the web specific components are handled by a single third party library, and as new web technologies, such as XForms, are likely to emerge in near future, all that the programmer needs to do is to upgrade the underlying Wt library and all the new features are available. Being a native C++ application means that the system comes with a small memory footprint and it performs faster than systems that rely on virtual machines. [43]

The programmer writing the application does not need to know whether the application is using the web interface or a native desktop interface as Wt programming interface provides same functionality as a native one, such as creating controls like buttons or combo boxes and assigning their events to callback functions. The same way, the library handles internally session identification and thread safety, so that multiple concurrent users can utilise the application simultaneously and the programmer does not need to take that into consideration.

The library comes with two different flavours of licences: free GNU General Public Licence that requires the source code of the library to be available if the application is distributed in binary form, and a commercial licence that comes with no such restrictions. Nevertheless, because the corporation has no plans for redistributing the library in binary form, a free version would be sufficient. [43]

Desktop user interface comes with several different possibilities but Windows Forms was chosen for it offers a convenient creation of panelled layouts and components that are strikingly similar to Wt. Windows Forms is a graphical user interface library that is included as a part of Microsoft .NET framework. Even though the user interface library is part of .NET framework, the programming language that is used to access the library is not restricted to only .NET languages and, for example, a C++ application can use the forms library with ease. The desktop user interface library is not a limiting factor in the same sense as web interface is, thus the actual implementation could very well be designed also using another user interface library, such as Qt or native Windows MFC (Microsoft Foundation Classes).

5.1.2 Logic tier in detail

Application engine that resides in logic tier is planned to provide core functionality of the entire application. The logic tier contains the actual configuration functionality and handles also creation of user interfaces using local databases that are included as part of the tier in earlier rough illustration of the system architecture. Even though, the databases do not actually qualify as being part of the application logic, the system layout drawing that was presented earlier was aimed to depict more simplified

component distribution of the application instead of strict division of multi-tier architecture.

The logic tier consists of four different elements, most notably the configurator engine that is used to compute crane variants. The structure of configurator engine cannot be fully covered in the scope of this thesis, but it is important to separate the actual configuration engine from the crane related knowledge and build it using fast and easily recyclable code. The configurator engine behaves like a software compiler that parses the product information out of the given input notation and returns valid order information or alternatively points out an error in the configuration. The configurator engine ought to be generic in terms that it should be able to configure any products regardless of their physical nature as long as the product can be described with the used rule language. Reason for keeping the configurator engine generic lies in the fact that the product platform is constantly increasing and widening, and fixing the configuration only to cranes would limit the usage options in the future.

Other important elements in the logic platform are the local databases and the synchronization service that handles all the information exchange to common global database over network connection. As the application has a need for offline functionality, local databases are used to contain persistent information of the application whenever network connection is not available. The local databases are synchronized to global databases by background synchronization service when a connection is available. The background synchronization service also downloads the software updates and handles all the connections to the external applications.

5.1.3 Database tier in detail

The responsibility of the global database tier is to work as a back-end server, database for configurator and to encapsulate all the connections to external systems. Basically, when the configurator wishes to access external application, it writes its information to the local database which gets synchronized to the global database. In example, when connecting to the DAS server for a crane drawing, the parameters of the DAS drawing are written to a local database and the synchronization service sends them to the global database, which handles the actual connection to the DAS server. Once the drawing is finished, it is written to global database, where synchronization server picks it up and copies it back to the local database. Similar way, the offer and customer information are transferred between the configurator application and the external systems.

This architecture was chosen to simplify the information exchange between the external applications. When only one service handles the communication outside the logic tier,

the system can be kept simple, and building a secure and reliable communication channel will be easier. Other benefits come from the fact that adding connections between new information systems and the configurator does not require building yet another communication channel to logic tier, and also the new system does not need to be exposed to public network, as the information exchange is done between back-end server and the external system. Since the calculation does not take place on the back-end server, the server side implementation does not come with restrictions for operating systems or databases.

Back-end server has all the necessary connections to external systems, which can be implemented case specifically. Once, for example, the new CRM system is constructed, it can be hooked to back-end server using a new service application that synchronises the information between the central back-end database and the CRM system. Basically a unified back-end server allows to connect the sales configurator with any existing system, and all of those connections can be built as separate projects to keep the implementation and maintenance simple, and to reduce the dependency from a sole system provider. Same logic applies, for example, to reading the BOM information from the PDM system, since using suggested architecture would allow BOM reading to be implemented as a completely separate application.

Since the user interface construction is controlled from the logic tier, which on the other hand is connected to the back-end, the entire user interface could be build directly from the central database. The architecture allows a possibility that each user could have a profile in central database, which would also contain information about screen settings, such as visibility of certain user interface controls or layout. Having a user specific profile would also make it possible to generate exactly similar looking user interface when user switches between the standalone client application and the web-application. In addition to connections to external systems, the central database could easily serve as an update server of the application and provide, in example, controlled means for non-standard quotation handling.

5.2 Meeting key requirements

Presented architecture suggestion was designed to clarify the key requirements for the project, and to depict one possible way that they could be fulfilled. The first key requirement deals with having at least 2-3 different user interfaces for differently skilled users. Having each user associated with a personalised user account in the back-end database can be used to realize the requirement, since the user interfaces of given users are generated run time due to separation between logic tier and presentation tier.

The second key requirement that copes with having both online and offline versions is the foundation of the presented architecture. Local databases that physically reside in the logic layer contain enough buffered information that the application can be used offline with limited functionality, such as lacking of DAS drawings. Synchronization service provides automatically networking capabilities much in the same way that an email client handles caching of mailbox whenever network connection is not available. The web-based version of the architecture utilises the same application base as the client version, except from the fact that the application is running on a dedicated application server. Since the differences between the two different versions reside only in the presentation tier, the user interfaces that originate from the logic tier would look similar between the versions, as only their presentation mechanisms would be different. Since the application server is also running the synchronization service and local databases in the same way that the client version does, it also makes the system tolerable for networking errors in database server end. The system distribution between the different computers is presented in Figure 12.

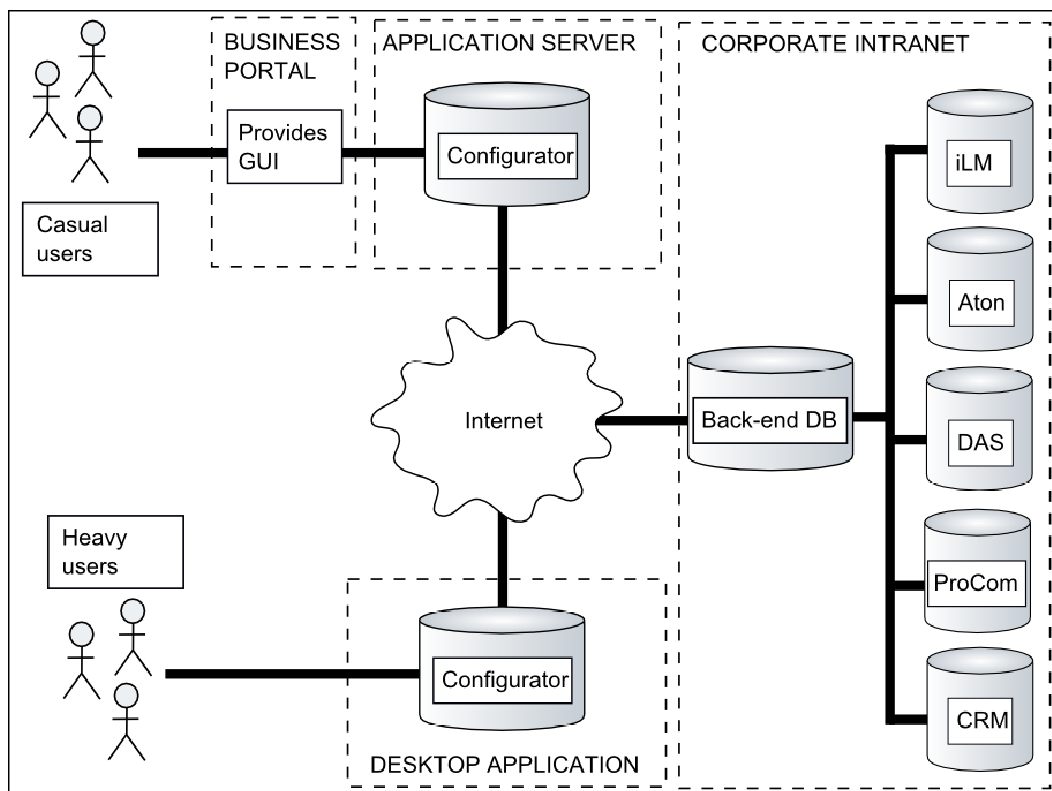


Figure 12. System distribution of the architecture suggestion.

The third key requirement dealt with integrations to other systems. In the architecture suggestion, all the external integrations utilise a common database server to handle the connections. The actual main application of the configurator system sends the

information to the back-end server, from where the information is then moved to external systems. This solution keeps the application logic and external systems separated, and also splits the project into smaller pieces for better maintainability.

The fourth key requirement of automatic sales BOM reading is direct continuation to third one, since the central back-end database is intended to contain also the generic product model information, and the reading of product information into the database can be done using a separate application thus reducing the complexity of the configurator. The product knowledge and sales BOM structure could be maintained in the current PDM system, from where a separate application could copy the information to the global back-end database. Another solution would be reading the sales BOM information into the back-end database directly from the database containing the engineering BOM structure but this implementation might turn out to be very complex achieve.

The fifth key requirement states that multibranding ought to be on a better level than it currently is. This issue was not particularly answered earlier in the architecture suggestion but the statement refers to both, look and feel of the application and the documents that the configurator produces. The intuitive method for solving the issue in case of the documents would be implementing a documentation engine to logic layer that would write content to specifically crafted template documents. The template documents could be regular formatted text files that are maintained in the PDM system that also contain specific content variable keywords that get replaced with the actual case specific text content in the configurator. The look and feel of the application can be controlled using style sheets in the web-version and a similar custom approach in the standalone version. The style layout can be once again stored in global back-end server from where they are synchronized to local machine according to user specific settings.

The sixth key requirement covers a wide range of different aspects of maintainability of the configurator. The maintainability aspect does not only cover the maintenance of the software but also up-keeping of the configurable product family. The application maintainability is answered by dividing the architectural suggestion into small mutually independent and replaceable components that can be easily changed one-by-one without changing the entire system. For example, if new needs for external connections emerge, those connections can be built to back-end server without touching the application tier. Maintainability of the product model depends on the actual implementation, since if the sales BOM can be read directly straight from the PDM system it would also offer the most natural system for maintaining product knowledge.

The seventh key requirement comes with a demand that the technology that is used behind the sales configurator should not get old very quickly. This architectural suggestion in terms of the configurator engine is based on C++, which is a programming language that is not likely to disappear from the face of the earth for a while. The language already has more than two decades of history; it has ISO/IEC standard ratification, and it is regarded as being one of the most compatible and fastest performing programming languages that exists. Although the configurator core is designed to be written in C++ in this suggestion, the selection does not limit the programming language of any other components, such as presentation layer or database back-end server.

The presentation layer can very well be written using different languages for different case specific purposes, such as C# for Windows desktop environment, or Java for another environments. C++ was chosen primarily for the reason that it is able to function together with most of the other programming languages and operating environments in a simple manner, so writing the application core with it would not shut any doors in the future, if new technology requirements come about. For example, if a new demand would rise, that the sales configurator must function on handheld computers or mobile phones, the application core would most likely fit into new environment without any modification, and only a new presentation layer for the new system would be needed. Again, this would also be the situation when new web-technologies are born: new technologies can be harnessed just by writing a new presentation interface.

The eighth key requirement deals with integration to Konecranes business portal. This requirement applies to the web-version, which should be made an integral part of the portal. This requirement is, once again, addressed in the separation of presentation layer from the application logic. As the physical location of the user interface layer is not tied to the location of application logic, there are no technical limitations that would prevent the web interface working in WebSphere portal environment. It is possible to run the presentation tier on the same application server that the application logic is running, which is somewhat a requirement if suggested Wt AJAX framework is used. Web based user interface could also be built on top of the application layer using more traditional Java based windowing that could run physically on a business portal. Depending on the business portal implementation, the AJAX framework based windowing could also be built so that it visually appears to be running on WebSphere portal, although it would actually be running on the application server. Having the entire configurator software to run on a business portal is, however, impossible in this architecture suggestion.

The ninth key requirement deals with reusing old crane specific calculation modules. The calculation modules are merged into the application layer within this solution suggestion. The calculation modules, however, must be isolated from the configurator engine to function with the smallest amount of dependencies elsewhere. The calculation components are the sole limiting factor in this architectural suggestion, as they require the application tier to run in Ms Windows operating environment and their isolation would be a sensible thing for maintenance purposes. Even though the calculation modules are intended to be kept intact in the future sales configurator, this architecture suggestion strongly recommends them to be rewritten in the future using a standardized technique that can be utilised also in the far future. Given that the suggestion advocates isolating the modules, they can be replaced afterwards one-by-one, once the resources permit.

6 ALTERNATIVE SOLUTIONS FOR NEW SALES SYSTEM

This chapter presents different alternatives for the sales configurator software providers that were selected for this project as possible third party software vendors. The chapter begins with a small market summary of sales configurator vendors, which explains how the different vendors made it up to the further evaluation and gives a general overview of the market situation. Each of the presented system providers in this chapter expressed their willingness for creating new next generation sales configurator and their solution proposals were subjected for further evaluation. Different companies had different approaches for the configurator solution: some of the companies offered a tailored custom-built solutions, whereas other companies had an existing sales configurator product platform that could be modified to meet the demands of Konecranes Corporation.

6.1 Sales configurator markets

Gartner research is an information technology research and analysis company that offers IT-related advices for large corporations and government agencies by request. Gartner performed a market scope analysis of sales configuration vendors at the end of 2007 and this chapter is based on it. Although the sales configuration market scope analysis by Gartner did manage to evaluate the strengths of different vendors, the evaluation metrics did not fully match the needs of Konecranes, which is why the results of the analysis are not taken as granted, instead the results are used only to point direction when looking for case specific best-of-breed vendor. [49]

The increased competition in the sales configuration market has divided the configurator providers into two different segments: large vendors that offer comprehensive sales solutions such as Oracle or SAP, and smaller vendors that aim to provide more customised best-of-breed functionality which are tailored to the needs of the customer and fit to the existing enterprise systems. The innovation and research in the field has continued, and research for new functionality, such as support for reconfiguration of existing and already installed commodities, is intense. The sales configurators are moving towards to support more complex product structures and business processes, although the analysis still points out that customers that are seeking a sales configurator for engineering-to-order processes should still carefully evaluate different vendors, since even the offerings of large suite vendors have gaps in offer-to-product chain. [49]

Lately, smaller niche players in the field have challenged large vendors by offering products that fit to existing information systems and specialize in certain distinct configuration challenges, whereas the benefits of larger solutions become more apparent when the existing ERP or CRM environments are provided by the same company. The future trends and research topics come in many, the previously mentioned support for reconfiguration being one of them. Many configurator customers are going through their third or fourth generation of sales configurators, thus the demands for ability to re-open old configuration models in new configurator have increased. Another future trend is an ever increasing demand for intuitive user interfaces for product knowledge maintenance, as corporations have a desire to have their own end users create and manage configuration models and processes without additional IT support. The trend for outsourcing the IT support is not only limited to the product knowledge update, as more than a half of the configurator vendors are offering deployment options where the configurators are being run on dedicated servers of software vendor and the customer only purchases the right to use the software. This deployment methodology, referred to as SaaS (Software As A Service), has been on the rise recently, although many industries still prefer possibility of having offline mode for situations with limited connectivity, which is why most of the configurator vendors also offer conventional on-premise licensing models. [49]

Evaluation criteria that Gartner used to assess different vendors consisted of four different main aspects:

1. *Overall viability* included the assessment of organisation's financial health, the financial and practical success and the likelihood of the company to keep investing in the sales configuration product.
2. *Customer experience* evaluated relationships that the customers had with the vendor, and the ways that they received technical, implementation or account support, and the availability of user groups and service-level agreements.
3. *Geographic strategy* judged vendor's strategies to direct resources and offerings to geographic regions outside the home markets directly or through partners.
4. *Product/Service* measured core goods and services offered by the vendor that compete in sales configurator market. The measurement included current product/service capabilities, quality, feature sets and the technology behind the software.

Because Konecranes is looking for a tailored best-of-the-breed solution, the preference criteria that Gartner used in evaluation cannot be directly assigned to the evaluation standards of Konecranes. The analysis emphasized the importance of viability of the vendor that offered the solution, which is also an important factor for Konecranes, although the financial viability measurement favoured large enterprises over smaller

ones for their larger financial success. When looking for a case specific optimal solution, the larger vendors may not automatically indicate better solution as some of the smaller players may even be more willing to tailor their products to the needs of the customer. Customer experience had also high weighting in evaluation criteria and even though the number of successful live production deployments was picked as a key metric, the evaluation also incorporated availability of different service level agreements and customer support programs. Geographic strategy has little importance for Konecranes point of view, as long as the vendor is able to serve the company in Scandinavian region, and the analysis by Gartner seemed to put more weight to vendors' presence in North American market. Product evaluation also included metric that evaluated how many different manufacturing styles, such as Ship-To-Order, Assemble-To-Order or Engineering-To-Order could be fulfilled with the same sales configurator. This, again, has little importance to Konecranes, since the corporation operates in Engineering-To-Order basis, and also a configurator that would be specifically designed for this particular manufacturing style might actually even prove out to be better than a one that is designed to serve many different styles.

Even though the criteria was not fully compatible with the needs of Konecranes, it still managed to point out obvious problems with some vendors, such as turbulent or uncertain future of certain solutions. The configurator solutions that received "caution" rating from Gartner are left without further presentation and they are: Firepond, Infor (formerly known as Baan), Oracle-JD Edwards, TDCI and Webcom. [49]

Brief results of market analysis are listed below:

- *Access Commerce* received positive rating for its increasing customer base. The configurator was seen as a good fit for complex Engineering-To-Order related organisations and received positive comments for intuitive user interface, support for both online and offline operation, and out-of-the-box Ajax support. The configurator provides licensing options that enable customers to choose between a traditional licence and externally hosted deployment.
- *BigMachines* received also good rating for its increased customer base but received criticism for operating mainly in a rather narrow sub-manufacturing segment. BigMachines configurator is offered through a hosted model and it does not incorporate on-premise version at all, although it offers the ability for clients to host the Web application in-house, with the vendor delivering the managed services on-site. Although the configurator is offered through hosted model, it also contains functionality for both offline and online use, and received positive remarks about Web services integration and improved single-sign-on support.

- *Cincom* was rated positive for its support for online and offline operation, intuitive user interface and quality of support. The configurator suits Engineering-To-Order manufacturing style particularly well due to its flexibility to handle calculations in the configuration process and being able to function together with third party ERP systems despite the fact that the company is also offering its own solution.
- *Oracle – E-Business Suite* got strong rating for its sales momentum, rules maintenance environment, ability to support many different manufacturing styles and large vertical use base. The benefits of Oracle EBS become apparent when used in conjunction with other Oracle applications as it offers deep integration with them. The company also has a large, global support team worldwide, as well as strong list of service partners. The configurator, however, does not provide standalone offline functionality although the distribution model includes both hosted and on-premises model.
- *Oracle Siebel* received a promising rating despite its uncertain future. The Siebel configurator benefits users mostly if it is used in conjunction with Siebel CRM and heterogeneous back office environment. The system features online and offline modes, and is noted to have very comprehensive functionality. According to the estimates of Gartner, the product line of Siebel configurator will be deprecated in near future in favour of Oracle Fusion Configurator, which includes changing the product configuration rules for the new system.
- *SAP* was the only large configurator vendor in the analysis that did not receive fully positive rating. Historically SAP has had two different sales configurators: Variant Configurator (VC) that comes with SAP R/3 package, and Internet Pricing and Configurator (IPC) which is part of SAP CRM 2005 package. Even though the strategic direction of SAP is to focus more on the Internet based version, both of the two versions are required to complete a front office to back office configure-to-order workflow. Another problem with the two different versions is that they are not mutually compatible and configuration models that are created in IPC cannot be replicated to R/3. Migrating the two different products together did not cause trouble to those customers that had existing in-house experience of ABAP (Advanced Business Application Programming) programming language, which is a programming language developed by SAP AG for its own purposes. Many SAP-based ERP customers were found using external best-of-the-breed sales configurators for reasons of flexibility.
- *Sterling Commerce* is another large player in the market, which entered the configurator market in first quarter of 2007 with its acquisition of Comergent Technologies. Sterling received positive ratings for the simplicity of its product maintenance environment and because of the complementary addition of

Sterling's assets. The configurator has an ability to handle complex sales configurations and the customer support was found to be good.

- *Tacton Systems* received good ratings for its ease of use even on complex configuration environments and notably also because of its installed customers rated the company's product and customer support highly. Tacton offers both online and offline solutions and received also positive remarks about simple-to-use environment for modeling product configuration rules. The negative remarks consisted only of limited visibility in North American market.

[49]

Gartner research provided several good aspects for further evaluation of the configurators and many of the configurators that were found promising in the research found their way for further review. Some of the larger vendors that were rated positive due to their sales momentum were left out for they were unable to meet the selected key requirements. Outside the Gartner analysis, several domestic and promising smaller vendors were subjected for further evaluation, as they appeared to be able to offer viable solution that would meet all the key requirements, even though they were too small players to show up in the international market analysis.

6.2 Methods for finding the alternatives

Different solution suggestions were found in many different ways. Some of the solution candidates had already an existing ties with Konecranes Corporation, whereas other vendors were found by following referrals in scientific articles, benchmarking other companies or simply by following sales configurator related publications, such as the market scope analysis. The number of different sales configurator providers is seemingly overwhelming but after excluding those configurator providers that only offer Internet-based electronic price catalogues from those configurator providers that actually offer a product design automaton grouped with artificial intelligence and complex engineering-to-order capabilities, the number of different providers was dropped dramatically.

Advanced sales configurators are subjects of scientific research of artificial intelligence and many promising alternatives had also academic background behind them. Different configurator providers went through an initial selection where their capabilities for handling the requirements were assessed subjectively before any of them were even contacted. The initial selection consisted of brief judgement whether the company is actually able to deliver a sales configurator that would satisfy all the requirements by looking at the references of previously completed solutions. Notably, all of the different

software vendors that made it through the initial assessment and to the pages of this thesis already showed their capability of being able to build a functional sales configurator that would be feasible for crane configuration and followed at least roughly the design guidelines that were presented earlier in the earlier literature review.

The initial selection was done by utilising previous experiment and tacit knowledge that the company had in software acquisitions, and consequently the measurement criteria also followed the guidelines of TIEKE (Finnish Information Society Development Centre), a non-profit organisation aiming to create viable tools and expertise for use in the information society in public and private sectors. The initial assessment criteria are presented in Table 1. [50]

Table 1. Initial assessment criteria for software suppliers

Criteria	Meaning of criteria
Feasibility of supplier and product	Is the product feasible for desired purpose?
Tailoring needs and possibilities of the product	Does the software need any tailoring to meet the requirements? How willing is the software supplier to tailor the product? How expensive will the tailoring be?
Flexibility and adaptability of the supplier and product to fit the existing information systems	Does to software fit together with the existing enterprise systems and architecture?
Time to market and availability of the software	How fast can we expect to get a finished product? Is the software finished and working?
Scalability of the software, extendibility in future	Can the software scale to yet unknown requirements in future? Is the software supplier willing to accept new requirements?
Maturity of software product and supplier	Can we expect the software provider to be able to keep its promises?
Research and development the supplier is making	Is the software supplier actively developing the application? Does the software solution have a future?
References and experiences of other customers	Have any other large enterprises adapted the application? Have they been content with the solution?

The different alternatives and software vendors that were selected for deeper review were evaluated by setting up meetings, where the requirements of Konecranes were presented as they were presented in the earlier chapters of this thesis. The software vendors were given time to prepare their own solution suggestions on how they would answer the given requirements, and other meetings were held after the initial one where the vendors presented their solution suggestions. All the different software providers and their solution suggestions are presented in following chapters in no particular order.

6.3 Solution suggestion by Wapice

Wapice Oy is a Finnish software company, originating from the city of Vaasa which offers customer tailored software solutions for industrial customers. The company has already existing ties with Konecranes Corporation as it has developed another sales configurator, chain system, which is used in the configuration of chain hoists and light lifting solutions. Other traditional industry customers include corporations such as ABB and Vacon. Wapice offers not only tailored enterprise application solutions, but also services from specification analysis to software support and is currently employing roughly 100 people. [51]

The company was selected as a potential software provider for a number of different reasons: they had previous knowledge of configuring cranes in the form of chain hoists and the employees of company also participated in scientific research relating to product variation, which led to understanding that the company has also a vision of future in sales configuration related areas. Also, the configurator solution followed academic separation between configured knowledge and software knowledge, which was covered earlier in the literature review.

Solution suggestion of Wapice is based on a tailored version of Summium sales configurator platform that the company is selling to various industrial customers. Wapice shared the opinion that it is highly unlikely that any existing configurator would satisfy all the demands of Konecranes Corporation, thus they expressed their willingness to adjust the Summium configurator platform to meet the distinctive needs. Summium is a sales configurator suite that includes also a number of different tools in addition to the configuration engine, such as tools for order, offer and customer management. The system is written in Java programming language and offers both standalone and web-based functionality together with compatibility features for integration to other enterprise applications, such as ERP, CRM and CAD systems. [51]

The role of Summium in the platform suggestion would be to act as a generic configurator platform and the contents of the application would not include any crane specific information; instead the crane related knowledge would be implemented externally using internal product description rules. The external calculation DLL modules would be embedded to the platform seamlessly using JNI (Java Native Interface) programming interface, which allows Java applications to access software components that are compiled to native binaries. Otherwise the system follows traditional three-tier architecture of J2EE based application.

Basically Summium configurator platform has one centralised server application that is built using Java servlet technology, which is connected to a database using XML based messages and HTTP protocol. Summium comes with a Java desktop client that communicates with the server, and which is operable in offline mode as well. When the application is started, information is replicated from the Summium server to a database residing at the local machine. Once the replication is complete, the Java desktop client can be used in offline mode. The contents of the local offline database are again exported back to the global database the next time the application is started in online mode. In addition to the desktop client Summium also has a web client, which runs inside a web browser and utilises HTML (Hypertext Markup Language) interface. [51]

Wapice representatives also noted that they have already identified and acknowledged certain problems in their current Summium architecture that they are on their way fixing. The current application architecture is user interface derived, which means that the user interface controls the core of the application and leads to a situation that every time when a user makes a selection, each and every parameter is calculated on the screen again even through there wouldn't be need for that. They also stated that currently used DOM (Document Object Model) architecture inside the configurator core performs slowly and has a large memory footprint. [51]

Having acknowledged the problems by themselves, Wapice explained that they are currently underway of changing the entire configurator architecture, and thus the future solution may differ slightly from the current Summium architecture. In future, the configurator core is designed to be intelligent and it is able to update the component selection information to a product model, when a value of one parameter is changed so that in future, the core steers the functionality of the user interface, instead the other way round that it currently works. The expectations towards speed and memory footprint are optimistic for future release, and the changed architecture should also have a better compatibility to a number of different types of upcoming user interface techniques. [51]

Another change for the future architecture of Summium is the communication with external services, which is planned to function so that calls to external applications can be included into configurator framework. The calls to external applications can be triggered automatically when a parameter is changed in the user interface, or simply by creating a push button that makes the external call. The built-in external interface can be used in communication with different applications, even those that are running in completely different environment and the same interface is also planned to handle the calling of crane calculation modules. The crane calculation modules are wrapped inside .NET components, which are accessed from Java application using JNI and C++ layers. [51]

Solution suggestion also aimed to answer the nine identified key requirements and the comparison of key requirements and solution suggestion is listed below:

- *The first key requirement* for having different user interfaces to differently skilled users was not seen as a problem, since Summium builds user interfaces dynamically according to user privileges. The users can also be grouped into different categories and build the views accordingly.
- *The second requirement* stated that the system should have both offline and web-versions. Summium does come with both versions and the standalone client can also be used without network connection. The two different versions do not look like each other for the time being but can be modified to do so.
- *The third requirement* dealt with integrations to other information systems. The chain hoist configurator that uses the same architecture platform already has connections to iLM and Pro systems, so there is a reason to believe that these won't pose a problem in new configurator design either.
- *The fourth requirement* of reading sales BOM information automatically from the PDM system could not be evaluated deeply, as the exact information that needs to be read is not thoroughly specified. Wapice representatives stated that they have done similar linking to other PDM systems earlier and assured that it would most probably work out.
- *The fifth requirement* stated that multibranding support ought to be better than in old configurator, and although the topic was discussed, it raised no particular concerns.
- *The sixth requirement* for easiness of maintenance is answered in the Summium so that the product knowledge maintenance is done using a rule engine for configuration. Each rule corresponds to crane selection options so that the updating product model does not require altering the configuration application. From Konecranes' point of view this would ease up the maintenance since Wapice would take care of implementing the configurator part of the

application, and it would only leave product model information and calculation components to Konecranes' responsibility.

- *The seventh requirement* dealt with having a technology solution that does not age very fast. Wapice representatives assured that J2EE based solution is likely to last long and the product family structure is stored in XML format, which is an established standard for presentation of structured information.
- *The eighth requirement* concerned integrating the web version into company wide business portal, which should not pose a problem, since J2EE technology is supported in WebSphere environment. However, the calculation components (DLL components) cannot run in the portal environment. The solution could be running configurator on a separate server and use SOA (Service Oriented Architecture) protocols for embedding the configurator into portal environment.
- *The ninth requirement* was a request of being able to merge existing calculation modules into the sales configurator. The requirement is taken into account in solution proposal by introducing a wrapper framework that is able to contain ActiveX based calculation components, which can be called from within the configurator engine.

[51]

It is safe to assume that the solution suggestion by Wapice managed to meet all the key requirements. The development time cycle that was promised was one year, provided that Konecranes is able to deliver detailed specifications of calculation modules that are needed to be added to the project with working example calculations, and that the BOM information from the PDM system is defined in schedule.

6.4 Solution suggestion by NSD Consulting

NSD Consulting is the company behind the current Markman sales configurator, established in 1993. The company offers information technology services and tailored solutions for industrial customers and is nowadays part of a larger Plenware group. The company offers software development and expert services on contract basis, so that the assignment takes usually place at client's site and supervision. The company offers services that vary from consulting to turnkey solutions and employs currently more than 50 people. [52]

NSD Consulting is originally a spin-off company from Konecranes sales configurator development group, and was selected as a potential software provider for the reason that and the company is thus likely to have the best initial understanding of products and problem domain. The previous configurator has performed in laudable manner and

many features of the old configurator are requested to exist in the new system. Also the calculation modules are initially designed to work together with the old Markman, which increases the potential of NSD for it helps to mitigate the risks of misunderstanding in requirements specification.

The solution proposal for the sales configurator is based on a completely custom made configurator platform that is build exactly to the demands of Konecranes. The proposal consists of recommendations of how NSD would build the system so that all the key requirements could be fulfilled. The technology recommendation addresses the base technology, software architecture and development methods.

The solution proposal suggests that the base technology that is used in the configurator would take advantage of WPF (Windows Presentation Foundation) technology. WPF is a graphical subsystem that offers a possibility to create applications with user interfaces that work as a standalone application or as a web-application utilising the same source code, much in the same way as Wt toolkit works as introduced in Chapter 5. WPF is a technology that is created by Microsoft Corporation as an integral part of .NET application framework. WPF applications can be created just as regular applications using Microsoft Visual Studio 2008 application development environment. The entire web functionality is handled automatically in the development environment, allowing the software developers design user interfaces in visual editor and not to worry about the web specific implementation since the same application can be compiled into a web application or a standalone desktop application. [52]

The reasoning why NSD chose to support WPF technology in their recommendation is that the same source code can be used in web and standalone applications, and it also represents the latest technology in Windows application development suggesting that the system has a long life-span and it works seamlessly also in future, being a standard development technology for Vista, for example. WPF technology per se is based on XAML (Extensible Application Markup Language) language, which is a XML based openly specified user interface markup language that is used in .NET framework. The description language is used to depict the user interface objects, such as push buttons and menus, and their binding into events and signals. When the application is run, the description language based user interface can be then rendered on screen using either web browser or the native user interface of the operating system. [52, 53]

The solution architecture in the application level consists of a plug-in framework model, where the application framework would be separated from the crane and product specific knowledge. The application framework would be responsible of taking care of all the core functionalities of the configurator application that are not specifically bound

to any product related knowledge. On the other hand, the product related knowledge would reside inside hot-swappable plug-ins, so that adding new product families would only require implementation of new plug-ins. The other way round, those common functionalities that are required from all the configurable products, such as drawing generation or unit conversions, can be embedded to the configurator framework. [52]

WPF application development requires Visual Studio development environment, and the technology recommendation suggests that the language used in the implementation would be C#, which is the most used language of the .NET based languages. The reasons that NSD listed for selecting C# are that .NET training in Finnish institutes uses mainly C#, which indicates that qualified employees can be found more easily. Another reason used to justify the language selection is that C# is considered, by NSD, to be a programming language of the future. This consideration is backed up with statistics from O'Reilly Media, a computer science books publisher that lists annual book sales, categorised by different programming languages. The book sales statistics are often used to measure the popularity and future prospects of programming languages, since they reflect those languages that the different institutes are teaching to students, and which languages seasoned professionals consider worth learning. The statistics show that the future prospects of C# look promising, and that its user-base trend is increasing, whereas the trend of other competitors is downward moving. Five-year trend of programming language books sold by O'Reilly Media is depicted in Figure 13. The figure separates C# from the rest of the .NET languages, although technically C# requires .NET framework to run on Windows. [52, 54]

The technology recommendation also features suggestions for development methods and processes. NSD notes that the most critical factor separating successful software projects from failures is the development process that is used, and that a good development process reveals problems in problem definition and risk mitigation, should there exist any. The development process recommended for the project is Scrum, an agile method that is designed to adapt to rapid changes and has high emphasis in quality. Agile development methods are designed to minimize the risks involved in application development by turning the traditional development cycle into smaller iterations. Scrum method is a process skeleton where the product owner and stakeholders create a simple list of features called product backlog together with the team leader, called scrum master. The scrum master then steers the programming team through development iterations, sprints, which usually last around one month. The idea is that during each sprint, the programming team creates new increment to the software in such a way that after each sprint the application is in a potentially shippable condition. This ensures that the software stays always usable even during the

development time, and the development team is also always directed towards a clear target situation. [52]

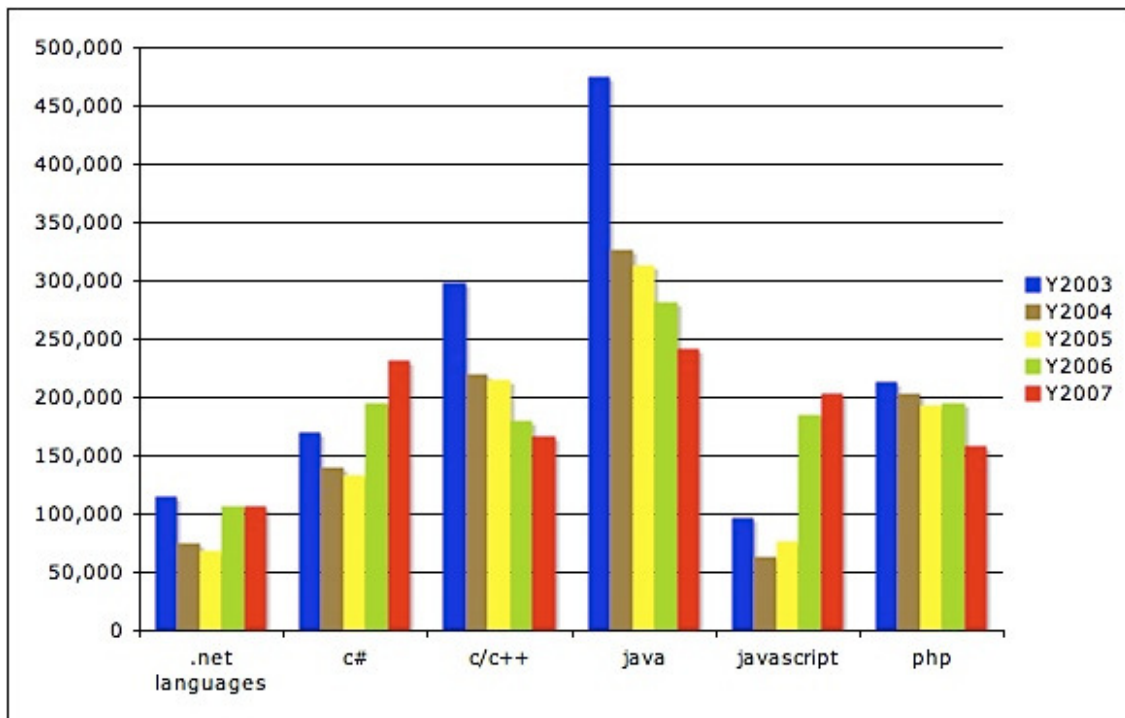


Figure 13. Market share of programming language books sold by O'Reilly Media [54]

Since the solution suggestion by NSD Consulting is based on a complete rewrite of Markman sales configurator it is safe to assume that the solution suggestion meets all the key requirements. The key requirements were noted in the solution proposal as acknowledged, and the recommendation also listed several other requirements that were extracted from other parts of the problem definition. The time frame that was estimated for the project was six months to initial version and two years to the point where all the defined functionality is implemented. [52]

6.5 Solution suggestion by Tacton Systems

Tacton is a global vendor of sales configuration software and services, with headquarters in Stockholm. Tacton is originally a spin-off company from the Swedish Institute of Computer Science, where the development of Tacton configurator began in 1992, preceded by many years of research and application development of knowledge based systems, logic programming, and configuration. The company serves several notable industrial customers such as GE Healthcare, Ericsson and ABB, and the configurator has been used for various different applications such as automatic

configuration of telecommunications equipment, pulp and paper machines and conveyor lines. [55]

When the initial skimming for possible software vendors was made Tacton was already noticed as one potential configurator provider for their vast references, and also for the fact that Tacton sales configurators are represented in this business region by the same software company, Modultek, that has built the PDM system used at Konecranes. Tacton Systems announced a strategic partnership with Modultek back in 2004 with the goal of bringing the sales configurator system closer to the PDM system offered by Modultek. This led to understanding that the Tacton sales configurator would work well together with existing information systems and product families. [55, 56]

Tacton sales configurator is a software suite, which consists of several applications; most important being the core of configurator engine, named TCserver (Tacton Configurator Server). TCserver is a back-end configuration engine, which can be embedded into any application that uses standard APIs, allowing also possibility for building own customized configurator application. TCserver is also included to other applications belonging to this same solution proposal that are TCsite and TCnomad. TCsite is a web based sales configurator that runs TCserver on the background, while TCnomad is a standalone offline application that runs the configurator engine locally. In addition to offline and online configurators, the package comes with an application called TCstudio that can be used to update and manage product family information. [55]

TCserver provides the core for configuration application and works independently from the rest of the applications. It is a separated configuration engine that is able to run on various platforms, such as Windows, UNIX, AIX, Sun Solaris or Linux. The engine per se does not offer any user interface; instead it can be embedded into various different projects and applications that provide the means for controlling the engine. Since TCserver is an integral part of the configurator suite offered by Tacton, the solution proposal also makes it possible to implement own custom made additions to configurator using techniques such as XML, Java, J2EE, .NET or COM/DCOM. The engine comes with all the needed functionality for configuration, and features also support for reconfiguration and ability to find optimal configuration according to any defined attribute or cost function, in example, optimization of a crane variant by weight or price. The engine incorporates a built-in functionality for scalable load balancing for multiple processors, so that the speed of the system can be increased by adding new hardware. TCserver is able to call external calculation modules during the configuration phase and offers also a possibility to divide the configuration into several steps in such a way that the external calculation modules can be called between the steps. [55]

Since TCserver does not include user interface itself, the other part of the solution proposal includes ready-made applications that wrap up the configuration engine into more usable form. TCnomad is a standalone application that is built to provide desktop user interface for configurator engine. It includes features to store and re-use the configurations and it is able to display 3D illustration of configured product with native Windows user interface. The application synchronizes the information with an online application server when network connection is available but is also able to function offline when connection is limited. [55]

In addition to standalone configurator application, the solution proposal also includes a web version named TCsite. TCsite provides the server end of the application together with web browser functionality for the configuration. Same way as TCnomad, the web version acts as a wrapper for the configuration engine, providing user interfaces and required functionality for price handling, quote generation, workflow management and user handling. The web version can be embedded into different portal environments including WebSphere environment, which is used for Konecranes business portal, and it supports general portal-oriented functionality such as possibility to use external user authorization handling for single sign-on operation. [55]

Although the solution suggestion seemingly relies on off-the-shelf solutions, the main idea is that the applications that use the configurator engine, TCserver, are tailored to the needs of the customer although the engine that runs on the background remains the same for all. Tacton builds the user interfaces around the configurator engine case specifically, reusing components from existing implementations of TCsite and TCnomad. The applications that are built around the engine are also the ones that are responsible of communication with the external systems, provide the caching and information synchronization mechanisms, and include subsystem for template based document generation. The product data and component inventory information can be read directly from Aton PDM system, while the solution proposal included also possibilities for reading price and customer information from other enterprise applications. [55]

The product data information is maintained with another tool called TCstudio, which belongs to the solution proposal package. TCstudio is an application that is used to administrate product structure information within the configurator, and it offers an editor for writing product definition rules that form the configurable product. The rule language is designed for non-programmers and Tacton considers it being one of the strengths of the solution, as the roots of Tacton configurator goes to academic study of depicting the product structure without any need of programming skills. The configuration engine uses constraint satisfaction approach to depict the configuration

rules and according to presentation of Tacton, the rule language is designed to limit the number of rules to minimum. As a case example of this, Tacton presented that ABB motors managed to reduce their number of rules from 80000 to 150 while increasing the product range 4 times. [55]

Although the sales configurator presentation of Tacton did not go very deep into technical details, the solution proposal included many case examples of different large-scale industrial corporations that had had similar needs and requirements than Konecranes. All of the key requirements were traced back and they were verified to exist in the solution proposal. Tacton was also able to present case examples of each different key requirement that were earlier implemented for some other company, thus it is safe to assume that Tacton manages to comply with all the requirements, and even though the company offers ready-made sales configurator solutions, they expressed their willingness to provide consultation and assistance in the product modelling and other phases of implementation, such as ActiveX DLL isolation or system interface development. [55]

6.6 Solution suggestion by Oracle Siebel

Oracle Corporation is an international software company, specialised in enterprise software products, that was selected to represent a large configurator vendor. For the time being, Oracle offers three different sales configurators: a lightweight solution JD Edwards, E-Business suite EBS, and Siebel configurator. From the three different configurators, Siebel configurator was chosen for further review due to a number of reasons. Siebel Systems was an independent software company known for their CRM solutions until 2005 when it became part of the Oracle Systems, and it is one of the software vendors that are being evaluated for the next CRM system. Merging the future sales configurator to the same software package could bring additional benefits not only in interoperability between the two applications but also in the cost and implementation. Out of the all configurator solutions that Oracle offers, Siebel was also chosen for the reason that, unlike EBS, it offers also possibility to use the configurator in offline mode.

The solution proposal by Oracle Siebel was compared against the key requirements and for the first requirement of different user interfaces for differently skilled users Oracle saw no problems in fulfilling the demand, since the system allows administrator to set up specific user interfaces based on user groups and privileges. Siebel has three different types of user interface frameworks available, which are used respectively depending on the capabilities of the web browser that is used with the application. Despite the browser driven user interfaces, the system also includes offline capabilities

using a sub-system named Mobile Web Client, as the second key requirement obliges. The user interfaces between the two different deployment models, online and offline, look similar to each other. [57]

The third key requirement dealt with having integrations to other EAI systems, and Siebel offers numerous methods to achieve connections to external systems depending on the architecture used. Siebel offers EAI integration platform that deals with linking the system with external third party applications and offers ready-made interfaces supporting variety of industry standards, such as OLE DB, Sun Java, XML and IBM MQSeries. The EAI platform is based on service oriented architecture and allows creation of case specific connections to the outside systems making it possible to call the configurator externally. The fourth requirement of reading product knowledge directly from the PDM system falls to the same category of external connections, and for the purpose of reading product family information Siebel offers a possibility to import parent-child related structures from an external source. [57]

Multibranding was not seen as a problem in Siebel either, as all of the configurator templates are customisable and can be branded as required, covering the fifth requirement. The sixth requirement about the easiness of maintenance is covered by product model editing tools that do not require programming skills to operate, and which allow the product model to be extracted and loaded into other Siebel implementations, such as Test and QA modules. The eighth and ninth key requirements raised no particular concerns, as the application can be embedded to web portal, and the external calculation modules can be called using several different system interfaces. [57]

The seventh key requirement of technology that does not get old too quick, however, resulted in a lengthier discussion, since Siebel Systems was originally acquired few years ago by Oracle, which is now building a new next generation business application suite named Fusion aimed to replace the old application suites in future. In spite of the upcoming changes in the distant future, the company will continue to provide a lifetime support program for the acquired companies' products, which includes also Siebel software suite. Additionally Oracle also provides multiple upgrade paths and assistance in migrating to Fusion platform when time for the new system becomes topical. [57]

6.7 Solution suggestion by Perspectix

Perspectix is a Switzerland based company established in 1996. The sales configurator product, named P'X5, that the company offers has deep roots in engineering-to-order

solutions and has several notable references in the field of mechanical, plant and electro-technical engineering. The sales configurator has deep integration to Siemens TeamCenter PLM system and it utilises JTOpen technology natively in the background meaning that the entire product configuration can be done with full real-time 3d visualization of the configured product. [58]

Perspectix was chosen as a potential sales configurator for Konecranes is planning to move the product structure to Siemens TeamCenter PLM platform in the future, and P'X5 could offer benefits for it is ability to function seamlessly together with the system. Another benefit is that the new CAD platform that the corporation is switching into uses structured JT file format, also by Siemens, which is natively supported in the P'X5 solution. As Perspectix has a strong partnership with Siemens and it utilises technologies that are readily compatible with the future systems, P'X5 was considered to be able to offer migration benefits for future purposes as well as a completely different view towards sales configuration by providing full 3d planning capabilities.

P'X5 is a sales configurator that is built around JTOpen toolkit, a visualisation library for the JT file format, that is used by Siemens NX CAD platform. JT is a lightweight data format that does not only contain the CAD geometry information, but also a linking to the rest of the product structure that is associated with. As an example, a CAD file could contain design geometry for a hoist of a crane, but it would also contain a reference pointer and placeholder to another CAD file, which would contain geometry for a gearbox of the hoist. Altogether, the JT file format does not contain only the design information of one part of the final product, instead it forms a structured CAD object database that depict the entire product structure in 3d space. JTOpen toolkit is a library for the handling of structured JT file format targeted to the software developers. It is a C++ API library that provides means for accessing the information in the 3d object database and a viewport for the visual presentation of the content. [58]

The sales configurator of Perspectix works in slightly dissimilar manner to other sales configurators, for the fact that the entire product design is made in 3d environment by plugging different configuration options together. The sales configurator does provide the familiar functionality for easy configuration in the sense that the salesman is able to select features and functionality from the list boxes, but with the exception that the salesman is also able to build the finished product by dragging and dropping parts to the 3d screen of the application. The user interface with 3d configuration environment is presented in Figure 14. The 3d visualisation screen is able to interpret the part connectivity meta information that is stored in the JT file so, that the options that are added to the configuration model can only be snapped to such positions where they are designed to fit. [58]

The actual configuration part of the P’X5 is handled by XML based ontology, which forms the core of the product model information by defining the structure of configuration model, against which the actual configuration parameters are compared using pattern recognition methods. The entire product definition in terms of selectable parameters and user interfaces are modelled in XML based presentation format using a separate editor tool that is built on top of Eclipse application framework. The editor tool provides a graphical user interface for the maintenance of product rules, such as the constraints that depict mutually exclusive components. [58]

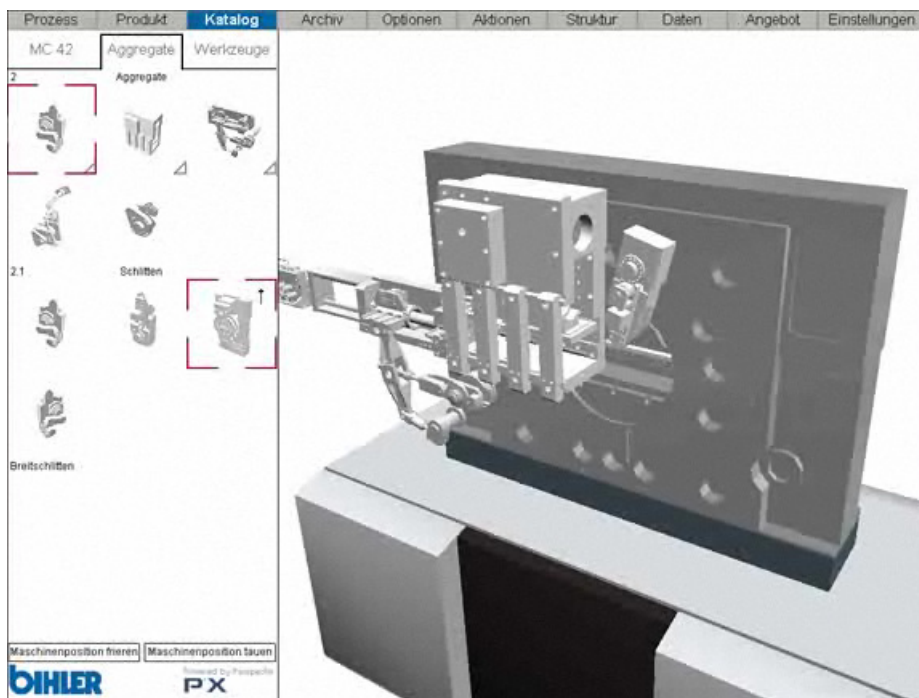


Figure 14. Visual 3d configuration in P’X5.

Perspectix offers a possibility for building different user interfaces for different users by utilising screen building editor that is embedded. The different user interfaces can be tied to the user privileges so that the different users can have different user interfaces, which was one of the key requirements. Another key requirement, the offline and web functionality was not seen as a problem either, as Perspectix offers also web version and the client version is able to operate in caching mode. The company offers two different versions for the web configuration needs, one that utilises AJAX and a separate browser plug-in for 3d presentation, and another one that runs the entire application remotely and displays the content of the application using a separate client plug-in. [58]

Integrations to other systems were not seen problematic, since the configurator uses natively XML based interfaces for all the information transfers and is designed to work with other system from the beginning, including business portal. The product family

information can be read automatically from TeamCenter PLM system, and the process can be implemented in such a way that PX'5 connects to the PLM system and downloads the JT database structure to a local workstation in an encrypted form. Multibranding and maintaining differently branded documents is done by directly editing XML based template documents, although the system offers graphical tools for user interface editing, which includes building rules that bring selectable options to combo boxes and prevent selecting the components that do not fit together. The core functionality is implemented in C++, and offers also a possibility for including calculation DLL modules into the application. [58]

It is safe to assume that PX'5 is able to meet all the key requirements for the next sales configurator of the corporation. Although, not all of the functionality can be directly compared to the other solution proposals, as the technological approach differs slightly from other rule engine based solutions. PX'5 offers also plethora of other features, such as performing real-time collision detection or designing the entire crane layout in 3d environment, which could not be objectively assessed since the needs of Konecranes were originally premeditated for the purposes of replacing the old sales configurator.

6.8 Solution suggestion by Cincom

Cincom is one of the oldest software vendors existing in today's markets, with a history dating back to late 1960's. The company is based in Cincinnati U.S., although it has several offices in Europe and it serves customers around the world. The company has a large number of industrial reference customers with solutions that vary from simple configuration problems into ones that require complex engineering efforts. The company offers configurator solutions that are designed to be simple and flexible. Cincom does not only offer sales configurators for its customers, but it also offers several other enterprise applications, such as ERP or PDM systems. Despite their own enterprise applications, the sales configurator is designed to work together with other corporate systems, developed by other companies, as well. [59]

Cincom was chosen as a potential sales configurator provider for a number of different reasons. The company received positive rating from the market analysis by Gartner research, and the sales configurator system that is offered is in effective use at one of Konecranes customers that has seemingly similar demands for complex configuration platform. As Cincom has a notable history with well-known customer references and it gained good remarks for its maintenance environment from Gartner, it was chosen for a deeper review and the company was asked to provide a solution suggestion. [59]

The solution suggestion by Cincom consists of two different parts: sales configurator, and a product configuration engine named Socrates, on top of which the sales system is built. Socrates is an application development engine developed by Cincom, which is particularly designed for the creation of knowledge-based software, and it provides core functionality for the configuration task among other features. Socrates is a hybrid engine that works in the background of the sales configurator and its maintenance tools, and instead of being solely a configurator engine it contains an entire knowledge environment for storing, maintaining and creation of configuration rules. The maintenance environment is designed so that it would allow non-programmers to create rule logic to depict the product structure utilising several different paradigms, such as selection rules, constraint satisfaction rules, generative configurations, pattern matching and scripting. [59]

Decision engine in Socrates handles diagnostic, selection, assessment, workflow and other similar configuration applications. Rules and decision trees are used to represent the knowledge in the decision engine, and the outcome is used to limit the number of available configuration attributes that can be selected by the user as the configuration proceeds. It is executed in the runtime to prevent incompatible selections during the configuration phase. The constraint engine is used when capturing the user requirements for complex products, and it is executed also in the runtime while the user is selecting required attributes, to remove those values that conflict with the existing selections. Hybrid engine also contains a generative configuration engine, a pattern-matching engine, and a scripting engine for product specification. Altogether, the configuration engine is able to handle vast amount of different configuration methods simultaneously to function in various different configuration environments. [59]

The sales configurator is a pre-defined application template that is built on top of the configuration engine, and tailored to the needs of a customer. The configurator is designed so that it can be modified to meet the specific requirements and it has several existing modules available, such as handling of configuration history, proposal management, customer and contact information, opportunity management, item configuration and catalogue, pricing with negotiation functionality, management of special options, and proposal generation. The configurator knowledge is maintained inside Socrates environment and the system is able to function in both offline and online modes, and it includes also a web-based version that can run inside a web browser. The sales configurator application is not intended to be a sole configurator; instead it is designed to cover entire offer chain from a quote generation to the final order. System infrastructure consists of a central database and a client application that is able to function temporarily without network connection. The product information can be imported to Socrates from external sources, such as ERP, CRM or PDM systems, and

the system is designed so that it is able to integrate external calculation applications or modules via several messaging interfaces, including COM+ that is required for calculation modules of Konecranes. [59]

Functionally the sales configurator consists of several modules. Opportunity management module provides features for managing offer quotations, and allows their creation, modification, status updates, tracking and projection. Quotation management module is responsible of integrating the contact information with offers and calculations, and it offers workflow functionality for the offer management. Solution configuration module is the actual configurator module, and it is used to build the product variant by utilising underlying Socrates engine, and a separate workflow management module handles the selling process supervision by acting as a centralised location that handles the communication and contacts between different key departments inside the organisation. Proposal management module is an editor that is used in the creation of custom tailored documents that the configurator produces. The module contains a repository of document element templates by default and allows new documents to be generated in a variety of output formats. [59]

Cincom sales configurator features were benchmarked against the key requirements of Konecranes, and it is safe to assume that the configurator is able to meet all of them. Cincom does not have representatives in Finland, but they saw no problems in handling the required service level remotely. Another suggestion from Cincom was to find a local software partner to handle on-site support in case one would be specifically needed. [59]

6.9 Solution suggestion by ACBIS / Epos

ACBIS is a sales configurator vendor that originates from an industry research project of University of Karlsruhe, Germany in the 1980's. The company has developed sales a configurator, named Epos, for more than 20 years and has gathered a collection of industrial clients, such as Iveco, Continental or Kögel, and currently the version number is already running at eighth. Epos configurator has existing ties to Konecranes Corporation, as a German subsidiary of the company, Stahl Cranes, uses the configurator as their primary sales tool.

The sales configurator solution integrates tightly with a CRM system called SalesManager, although it is able to function together with other CRM systems as well. The Epos sales configurator itself provides the configuration engine for the task, and in case of Stahl Cranes a third party software-consulting company, Infcon, provides the rest of the functionality. The configurator is specifically designed for machinery

industry markets and the design is based on the principle to easily create and maintain configuration rules without requiring programming or special IT-knowledge. [60]

Epos configurator engine consists of two different elements: user interface editor that is used to create all user interfaces and handle the inputs. The user interface editor is also be used for the localisation of the application and to create the restrictions that prevent selecting incompatible options in the configuration. Another part of the configuration engine is its underlying database, which contains the entire product model definition and the contents of user interface selection options. The configuration procedure works in such a way that whenever user selects an option from user interface, the database is queried for new options, which are updated on the screen thus preventing the wrong choices. [60]

Epos configurator was compared against the key requirements as follows:

- The first key requirement of multiple different user interfaces could be fulfilled as the user interface editor allows building separate screens for different users. The user interfaces need to be built with the editor, and they are not created automatically by configurator engine, which makes it possible to tune the visual appearance manually.
- The second key requirement for offline and web functionality was also fulfilled as the configurator works as a standalone application but includes also a web version. The web version development is moving towards utilising AJAX technology and a new version is in the design.
- The third requirement dealt with integrations to other enterprise applications. Epos configurator does not offer predefined functionality for all of the different systems, such as the PDM system, but however, as all the information in the configurator is stored in easily accessible format, it is possible to build additional applications that handle synchronization to other systems.
- The fourth key requirement was continuation to the third requirement and exacted product information to be read directly from the PDM or PLM systems. This functionality is not available but could be fulfilled by building a third party application to handle the database synchronisation between the sales configurator and product knowledge databases. All the product knowledge in Epos configurator is stored in local databases and it would be possible to create a separate application that can read the product information elsewhere.
- The fifth key requirement of multibranding could be achieved by crafting user interfaces for each brand with the help of the user interface editor. Each brand can have their own interfaces and documents, although all the different screens and documents must be recreated for each different brand separately.

- The sixth requirement dealt with having a technology that does not age very fast. Epos configurator is under active development and although the current technology is based on Visual Basic, a new version that utilises more up-to-date technology is under development.
- The eighth key requirement for integrating the configurator into business portal is not a problem, as the web version can be integrated to other environments, although fitting it into WebSphere portal requires some extra work.
- The ninth key requirement of utilising old calculation modules in the new system can be fulfilled, as the configurator system is able to call external DLL modules from the framework.

[60]

6.10 Other solution suggestions

Few promising sales configurator vendors were selected for deeper review even though the actual software platform was not able to provide the all key functionalities that was required for the project. The configurators were evaluated for they offered a different approach to the software problem and had some notably good characteristics despite that they lacked some other key functionality or service level potential. The deeper review was performed to widen up the perspective regarding the sales systems and to assess if the key requirements were too restrictive by nature, so that a different approach to the functionality would actually generate a better outcome for the final product.

Most potent of the sales configurator providers that came very close to match the key requirements was Variantum, which is an expert company that is specialised in product data management and mass-customisation solutions, established in 2002. The company employs five people and it is a spin-off company of Product Data Management Group, a research group of Helsinki University of Technology. The company was selected for deeper review for the development crew included the most of academic professionals out of all of the vendors, and some theory parts of this thesis are based on the publications created by the same individuals that have influenced the development of the configurator. [61]

The solution proposal included an implementation of a sales configurator engine excluding the advanced functionality for offer and order handling that are required from the new system. The proposal relied heavily on the implementation of the configuration engine and it was suggested that another software partner would be selected to implement the rest of the sales management functionality around the engine. The configurator engine per se contained advanced functionality for product configuration

and supported sophisticated mechanisms for product definition and automatic user interface generation according to the product definition. [61]

Another solution proposal that did not quite manage to match the criteria was an India-based company named Idola Fori, that offers software solutions which are tailored for customer demands. Idola Fori has previously made a crane sales configurator for another crane manufacturing company and was willing to offer similar solution for Konecranes as well. The sales system that Idola Fori offers is built on top of a CAD system and it works like a design automaton by utilising the underlying software. The configurator handles the steel structure optimisation by itself and builds the configurable products from a database of predefined 3D components. The configurator features both offline and standalone versions and it is able to output 2D and 3D drawings. The system does not, however, offer a separable configuration engine, instead the products are modelled statically using the underlying CAD application, and since the system is only designed for the configuration of cranes, adding new products such as service packages or lift trucks might turn out to be problematic. [62]

7 EVALUATING SOLUTIONS

Comparing the different sales configurator vendors objectively can be difficult, as all of the vendors claim superior capabilities of their products. The task of evaluating the relative supremacy between the different solution proposals is a tedious one, for each different solution proposals have strengths that are not directly comparable to other proposals. This chapter aims to find objective metrics to compare the different sales configurator solutions to maximise the benefits for Konecranes Corporation. The metrics presented here are biased to favour the exact needs of Konecranes and thus the outcome of this comparison does not reflect the general truth about the superiority of a certain configurator over another, which may be completely different in a case of another company. Once the decision-making criteria are determined, the different solution proposals are evaluated against it, and the most feasible solutions for the next sales configurator are selected.

7.1 Finding the best solution

Different sorts of methods exist in academic world for the evaluation of best software vendor for a given project, but most of the research focuses around the measurement methodology instead of dealing with the actual selection criteria. On the other hand, those studies that do focus on the selection criteria are oriented to a rather narrow area of evaluation, such as risk mitigation, financial aspects or quality ensuring. Basically the problem behind the final evaluation reduces to a multi-criteria decision analysis, which is a field of science that focuses on comparing the different alternatives in a neutral manner aiming to support the decision makers. The methods for decision making vary from fuzzy methods to different tree structures and their variants, although at simplest, a decision-making matrix can be used to score different features of the application and the different alternatives can be assessed by comparing the final points.

The decision-making matrix was chosen for this project as it provides good enough method for evaluating the alternatives and most of all it is intuitive to understand even for those that are not familiar with the method earlier. A decision matrix contains a list of different criterions that are given a weight factor that signifies the importance of the measured attribute. The different attributes need to have different weighting factors to allow a fair comparison, as different attributes have a different importance for the project. The weight factors also enable a possibility to perform sensitivity studies for the results, so that it is possible to see how the results change if different attributes are weighted differently. [63]

Belief decision matrix is an extended decision matrix, which is intended for dealing with those problems that have both quantitative and qualitative criteria, and the reasoning must be done under uncertain or random circumstances. Unlike in conventional decision matrix, the measured attributes are not given a single grade; instead the grade is replaced with a specific belief structure. The belief structure is a probability distribution, which reflects the assumptions that different reviewers have for the measured attribute so that the given grade consists of several different values that form a probability graph. In example a grade that would roughly be “good” in a normal decision-making matrix, could be expressed in more exact manner as 40% excellent, 50% good, and 10% bad, using a belief structure. The benefits of belief structures come apparent, when the true grade cannot be fully assigned, and the judgement must be based on assumptions. [63]

After a careful consideration, however, the belief structure as such was left out of this comparison. Even though utilising belief structures could provide more exact mean for assessing the criteria, it would also make the resulting matrices less intuitive to interpret and the actual benefits of utilising the more complicated methods are questionable, for many of those criterions that were considered difficult to assess were divided into smaller sub criterions that were distinctively inquired from the configurator vendor. The grades that were given to configurator vendors were determined to be decimal values instead of integer values to provide more continuous way of assessing the offerings.

The list of decision criteria was collected from various sources: scientific articles, professional statements and Internet articles. None of the source articles offered complete evaluation criteria that could have been used in this project in a form that they were presented in the articles; instead the different ideas were handpicked and altered to match the needs of Konecranes Corporation. Many source articles focused on a rather narrow sector of software evaluation, such as risk mitigation, or financial aspects. However, this project requires assessment of all of the different factors as an entity, which is why the different criterions were collected from various sources and eventually compiled into a complete list of 45 different attributes, each of which having a weighting factor to describe the importance of the attribute in the final evaluation. The weight factors were set by a number of IT-professionals and stakeholders working for the project, and the final weight factors were selected so that they would reflect the opinions of the majority.

The evaluation works so that each attribute is graded by giving it a score between one and five, which is then multiplied by the weighting factor. Eventually all the scores are summed up for a total grade, according which the different sales configurator vendors

are ranked. In the final ranking stage, the weight factors are still subjected for sensitivity studies to assess the reliability of the final result.

7.2 Evaluation criteria

Even though the evaluation criteria that was used by Gartner research was not fully feasible as such for the selection of best-of-the-breed solution, it still provided a number of good aspects that can be adapted to the requirements of Konecranes as well. Market analysis did put a lot of weight for the vendor strength and the future prospects of a configurator, which are all good criterions to include into this study. However, Gartner did put a lot of weight to the global vendor presence, the size of the company, and the support for multiple different manufacturing styles. Vendor presence does play an important part for Konecranes as well, nevertheless, the system administration, development, and maintenance is planned to take place in Finland, which is why a strong vendor presence in the U.S. markets does not bring much additional value for the deal.

The size of the company is an important aspect as well, as it brings sustainability for the configurator vendor, although, as even Gartner suggests, the bigger the software supplier gets, more rigid it becomes to adapt to the needs of a customer, which is why companies that have complicated demands should look for a best-of-the-breed solution. Corporate stability and future commitment to configurator development were picked as part of the evaluation criteria; nevertheless, the actual size of the vendor was left out from the list. The size of the vendor does reflect the ability of the software vendor to respond to critical situations and customisation needs. These criterions, with many others, were picked from the Gartner analysis into separately measured attributes. [49]

Another source for the evaluation criteria was a software purchase guide of TIEKE that was used already at the preliminary phase. The guide listed few evaluation criterions in a high level of abstraction, which were refined into more exact requirements to ease up the measurement. The criterions that were picked from the guide included features of the application, availability of maintenance, and the customisation capabilities. Several other articles were used in a same manner as the ones mentioned above, in example many scientific articles focused on software development risks from either project or product point-of-view, which influenced on criterions spanning from software features to vendor proximity. All the different evaluation metrics and their corresponding weight factors are presented in Appendix 7. [50]

The evaluation criteria are divided into 11 different categories that measure the different aspects of the software project. The first sub-category, entitled “Comparison to theory”, compares the solution suggestions to academic research of sales configurators, which was briefly covered in Chapter 3.1. Main emphasis in this section is in the separation of product knowledge from the software knowledge, but also a vendor having a constraint based configuration engine is considered as a definite advantage, since it was also seen as a future direction in the configurator development by Gartner analysts. The second category measures the characteristics of the solution proposal and puts a lot of weight to meeting all of the key requirements presented earlier. All the key requirements are mandatory for all of the configurator vendors, although the levels of punctuality how these are interpreted between the offerings vary, and this is measured by giving a better grade for solution proposals that are closest to the original requirements. Other highly weighted attributes in this category include usability and adaptability features. Usability grades are given by comparing the solution suggestions against common usability reference articles, such as Jacob Nielsen’s ‘Alert box’, and the adaptability, which focuses on the systems’ ability to adjust to rapidly changing IT-infrastructure, is graded by evaluating their architectural flexibility. [64]

The third category includes technology related evaluation criteria where compatibility with existing systems, openness of the system and the speed of the functionality among others were given high weight factors. Compatibility with existing systems is an obvious evaluation criterion, which is also one of the key requirements, and thus required from all of the vendor candidates. This criterion measures not only the compatibility to the existing systems but also how it is implemented, and if there are any particular strengths in the implementation that makes the connectivity faster or more reliable. Direct continuation to this metric is found further in the list where the amount of work required for fitting the configurator into other enterprise systems is measured. Openness of the system is regarded as a very important matter, as an ability to access the system externally provides safety for future prospects by emphasizing the fact that the configurator vendor does not need to hide the technology and even allows implementation of supplementary applications. Another criterion closely related to the openness measures whether the selected architecture ties hands to a single software provider in the future IT purchases.

Category four includes measurement criteria involving business stakeholders. Most weighted attributes involve speediness of how fast product definitions and pricing information can be updated to the system, although fitting the configurator into general IT-strategy plays an important part as well. The fifth category contains several highly weighted attributes that revolve around the services and support that belong to the part of the offering. The most favoured metrics are vendor proximity and the support that is

received after the initial version of the configurator is deployed. Vendor proximity is not meant to measure physical proximity of the software vendor; instead the main focus is in the evaluation of how willing the vendor is to commit to instant support in case of urgent or unexpected demands, and the desire to cooperate in such situations. Another criterion, after implementation support, relates to previous one, although this attribute is used to evaluate different service level agreements that the vendor is offering, if any.

The sixth category measures the costs that are involved in the project. The total cost to the first finished version did not get as big of a weight factor as a combined criterion of payback time and return of invest. Payback time and return of invest are calculated from different variables, that does not only include the licence price but also building the product models and the consulting that is needed to it. The licence price is included in the metric that evaluates the total cost to the first finished version of the configurator, however, in the long run it is expected that the building and maintenance of product families will eventually cost more than the licence fees, which is why the payback time is rather used as a financial metric. The maintenance of product family information brings remarkable part of the long-term costs to the project, thus modelling the product information in such a way that it can be easily maintained is a key factor in keeping the running costs low. This is also one of the founding reasons why consultation and support during the early implementation is vital for the sake of the total costs in this project.

The seventh category assesses the amount of customisation that is needed, and different means of doing it. The eighth and ninth category, on the other hand, are used to measure the strength and the future prospects of the corporation that is offering the solution. The eighth category focuses on the financial strength of the vendor, since the software is supposed to have a lifespan of more than a decade, it is only reasonable to select a software vendor that is likely to exist the entire expected life cycle. Other evaluation metrics in the same category aim to measure whether the vendor has enough qualified personnel to ensure the viability of the technology platform also in the future, and if the vendor has any existing customer references with similar requirements. The ninth category measures the vision that the vendor has for its own operation and its future direction, and notably also the versatility and the flexibility of the organisation. The tenth category contains evaluation criterions for risk mitigation and the ability of the vendor to respond to unexpected problems during the development phase. The risk mitigation category has only three different measurable attributes listed, since most of the features that contain risk analysis belong to other categories presented earlier. The eleventh category contains only one feature that could not be fitted into any other category and it measures whether the vendor is willing to offer software consulting also in other needs outside the configurator development.

Many attributes that are included in the list have overlapping with other attributes, thus the actual division into different categories is hazy. The overlapping between the different attributes, however, is intentional as they are sometimes meant to measure even same features of the configurator, although from a different point of view. The different categories also end up having an uneven importance in the final grading, as the amount of measured criteria and their weight factors vary between the categories.

7.3 Evaluating solutions

The final evaluation consists of grades from one to five that were given to different solution proposals according to prevailing knowledge of the solution suggestion. The grades were determined after a careful consideration in a meeting held with a configurator stakeholder group and all of the grades were given at once, after each of the configurator vendors were met. The grades that different solution proposals attained are presented in a matrix in Appendix 8, and the grading values that were given to vendors followed predetermined principles as follows:

- 0 – Feature does not exist
- 1 – Poor
- 2 – Worse than average
- 3 – Average
- 4 – Better than average
- 5 – Exceptionally good

The grading matrix consists of a table where different sales configurator vendors are organized in columns, each vendor having two distinct columns: one for the grade and another for a weighted grade, meaning the grade is multiplied by the weight factor. The evaluation criteria are listed on the rows of the matrix and their identification key corresponds to the evaluation table in Appendix 7. All of the vendors are referred in the evaluation matrix only as numbers and they are presented in random order to protect the interests of companies involved. The sales configurator vendors that made it to the final evaluation are the same ones that were more deeply presented in Chapter 6:

- Wapice Oy
- NSD Consulting
- Tacton Systems
- Oracle Siebel
- Perspectix PX'5
- Cincom

- Acbis / Epos
- Variantum
- Idola Fori

Initial impressions of all of the final contestants were promising, and surprisingly many of the configurator vendors took many features, such as offline functionality as granted. At the beginning of the project, the need for the offline functionality was questioned and raised heated conversations, however, the most of the sales configurator vendors, even the ones that did not end up to the final list, were offering offline functionality as a standard feature. The most difficult requirement to fulfil turned out to be the reading of product family information directly from the PDM system. Basically the actual reading did not pose a problem but extracting the sales BOM structure automatically from the engineering BOM structure did turn out to be something that none of the vendors were able to completely fulfil, thus the requirement was interpreted so that the product data information should be read from the PDM system, but a separate product family structure for sales purposes could be maintained along with the engineering BOM structure. One vendor, Perspectix PX'5, however promised that their system could handle the synchronization between the different product family structures automatically.

Many of the solution proposals also employed similar architectural approaches towards the solution as the theoretical part of the study suggested, meaning that most of the configurators had a separable rule engine that utilised same reasoning methods that were presented in the scientific articles. Rule based reasoning and constraint satisfaction solutions were the most common ones, and some those configurators that still used the rule based reasoning were planning a transformation towards constraint satisfaction engines.

7.4 Initial impressions of different architectures

Despite the fact that Wapice representatives themselves had identified few issues in their own platform, the solution suggestion left a rather positive impression. The issues they had noticed were already under inspection, though the most comforting part of the solution suggestion was the promise of service level that would be part of the package, where the Wapice representatives stated that they would do whatever adjustments that are needed for the project to succeed, even ones that do not directly involve the sales configurator. Architecturally the configurator solution followed the academic division of having a separate configurator engine and even the problems that they had identified in their current solution were rather small and already being fixed.

NSD Consulting had also a promising configurator suggestion, which in spite of lacking a rule engine based approach, still managed to offer flexibility in the product structure. The strengths of this solution suggestion are definitely in the existing knowledge that the company has of Markman and its calculation modules, which is why NSD managed to give the most accurate project plan out of all of the contestants. The downside of the solution suggestion is the lack of actual configuration engine, although the same issue is answered by separating the product knowledge into plug-in components.

Tacton Systems had a rather interesting solution in comparison with the other contestants. The configurator engine was a completely separable from the rest of the applications, and was designed in such a way that any third party developer could build applications around the configurator engine or embed it into other existing projects. The client applications are built in a same way as well: the company builds customer specific applications on top of the configurator engine, and utilises their existing software modules to cut down the development costs. The architecture left a positive feeling for the level of freedom that it offers. The architecture would allow Konecranes to develop supplementary modules for the configurator if needed, but also offer possibilities to purchase the work either from Tacton or any other software company. Basically this solution would help to minimise the risks involved in the configurator development and open up new possibilities, such as teaming up with other industrial partners that use the same configurator.

Perspectix PX'5 was a rather difficult to compare with the other solution suggestions for being so much different from the rest of them, and left mixed feelings because of that. PX'5, indeed, is capable for configuring very complex product structures by allowing the user manually drag and drop different parts to the 3d view on the configurator screen. The complexity, however, might be a little bit over the edge for the purpose of Konecranes, given that the salesmen that are meant to use the next sales configurator are not engineers, and the system felt as if it would suit a technical sales team better. Perspectix was the sole sales configurator vendor that was able to promise an answer for reading the sales BOM structure directly from the EBOM structure given that the EBOM model is built according to certain specifics.

Cincom configurator again employed a slightly different approach to the configuration by offering a complete rapid application development (RAD) environment that contained the configuration engine. The development environment based solution suggestion was an interesting one, although very hard to measure. Given from the case examples, the configurator is capable of fulfilling all the different demands and offers also very good functionality for mastering the entire sales process flow from the offer

stage to final production stage. Things that were hard to measure included the amount of work that would be required for building all the required functionality with the environment, and the amount of assistance that would be needed to finish the task. However, the suggestion left a feeling of a very capable system if properly implemented.

Oracle Siebel configurator is a sub-module of Siebel CRM system, which was also apparent from the configurator design, and undoubtedly the advantages of the configurator would become more clearly visible when used in conjunction with it. Oracle has a strong organization with plenty of expert knowledge behind it, although implementing the configurator solution would require quite much software consulting in order to be fitted to the key requirements and Oracle representatives recommended hiring a separate software company to do the job. On the other hand, a big system vendor does have strong industrial knowledge and is most likely able to offer a long-lasting solution with steady upgrades also in the future.

Acbis/Epos configurator was another one that left mixed feelings. On the other hand, the system showed capability for fulfilling all the different needs that are posed to the new configurator, nevertheless implementing all the features to the framework gave an impression of being a tedious task. Basically the idea of an UI editor did not differ very much from the RAD environment that Cincom offered, although having to build an user interface for each different purpose, separately from the constraint engine, left with the thought that the maintenance of a multibranded and multilingual system could turn out to be difficult. The biggest advantages in the solution was the integration it offered to the CRM system, and also the fact that the configurator has already proven to be capable of configuring cranes in Stahl Cranes subsidiary of Konecranes.

Variantum earned its way to final evaluation in spite of the fact that it missed few key requirements. The solution involved another software vendor taking care of the surrounding functionality, whilst Variantum would have taken care of the configurator engine development. This division, however, did leave concerns for it would require dividing the project and its specification among several parties, which could lead into new complicated situations. The configurator engine itself turned out to be flexible and suitable for complex configuration tasks. Idola Fori, on the other hand, raised concerns about the vendor proximity and how the actual project could be handled with a subcontractor that works in such a remote location without any nearby partners. In spite of very competitive pricing, several other aspect of the solution proposal, such as lack of specifics in the architecture design, made it quite difficult to grade.

7.5 Result of evaluation

None of the configurator vendors that were graded were particularly bad. According to grading, three sales configurators received remarkably higher results than the others involved in the scoring. The sensitivity studies performed to the result matrix revealed that the three winning configurators tied so close to each other that altering only few grades would affect their mutual order, thus all the three sales configurators were selected as winners of this evaluation and will be presented as a final result of this study. The three winning alternatives are subjected to one more final test after this study where some of the key requirements are tested to actually exist, by merging one of the calculation modules to the configurator framework and building one simplified configurable product in order to see that the system is actually capable of fulfilling the most crucial functionality.

The winning configurator entries were Tacton Systems, Wapice Oy, and Cincom that all came very close to each other in the evaluation. Tacton Systems eventually won for the flexibility and the levels of freedom that the architecture offers, although Wapice came very close due to the fact that the level of service offered got very high weighting in the evaluation criteria. Cincom on the other hand had a very solid offer in all the different areas and eventually scored among the winning solutions. What made Tacton Systems winner was that the architecture, where the configurator could be coupled together with any other application, offered so many possibilities for the development of new functionality, that it would not only offer viable solution for all the requirements but also certainty for future development due to openness of the system.

As stated earlier, none of the other configurator solutions were bad either, and the biggest reasons why some configurators received lower scores mainly rose from the fact that they were originally designed for slightly different purposes. An example of such a configurator would be Perspectix PX'5, which was technically among the best but would have served the needs of non-configurable special-crane design team better than the average salesmen of standard lifting business area, and for those companies that have a very complex product structure, requiring manual on-site design of the end-product, the solution would have been the best. The same logic of focusing to different purposes applied to most of the other configurator vendors as well.

8 CONCLUSIONS

As the configurator platform is tentatively selected, the next step in the project is to move forward and to begin the planning of the actual implementation phase. During this study many different details came up that were not properly premeditated in advance, and that need to be resolved prior the project can enter realization phase. This chapter proposes further actions that ought to be taken before planning the project and also sums up the findings for this thesis. The chapter also reviews the findings of this study and aims to provide a critical lookup backwards to assist in other projects of similar kind.

8.1 Further actions

As the initial selection for the most potential sales configurator vendors is now made, the project continues to the next phase where the winners will be presented to a board of IT-managers of the company, and after their final decision the project enters the planning and implementation phase.

The first thing that ought to take place in the planning stage is a complete requirement analysis that must be made together with the vendor. The requirements analysis will be based on the outline that was gathered in earlier stages of this study, although at the time when the vendor for the project is known, a more in-depth analysis can be made so that it includes all the different interfaces and information exchange environments. More requirements for the project will be collected as the requirements analysis takes place, and special innovation days for the final features of future sales configurator is already being planned. The innovation days are held with the assistance of a consulting company, and the purpose for the event is to find yet additional features that should be implemented to the system prior the final requirements are decided. The participants to innovation days are collected from the representatives of different user groups and brands to ensure that all of the opinions will be noticed in the requirements.

Another task that ought to be done prior entering the implementation phase is a formal mapping of the product and product family structures. One of the biggest problems during this assignment was to find a way to depict all the different needs that follow from the diverse product range for the vendor candidates so that they could understand the scope of this project. Having a model of a product family structure presented in some formal manner, such as an UML chart, would ease up the understanding of different limitations and requirements for the final configurator. Not only would the

visual product model help to understand the demands, but it would also allow running the project asynchronously so that, to speed up the project, the crane specialists could begin to work with the model while the actual configurator software could still be in the design stage. No matter how the GPF model is stored inside the sales configurator, an accurate representation of the model could also serve other purposes, such as easing up the maintenance of the product family, and a centralised knowledge repository would also fit to the corporate IT strategy aiming towards a centralised storage of business information.

As the project contains high risks, not only in the software development but also in business sense for the fact that configurator generates the crane design on the fly, a separate risk management evaluation ought to be made together with the selected configurator vendor. A complete risk management research must be performed together with the software vendor that addresses key points in the risk identification: test cases, methods for finding the problems, and guidelines for dealing with the unexpected situations. It is important to include the selected software vendor to participate the risk evaluation in order to ensure mutual understanding of the matters in question. Utilising a third party consultant to assist in the risk identification might be justified, as a professional coming from the outside of either of the organisations would most likely be able to pinpoint potential problems in a more accurate manner.

Calculation modules are specified to stay intact, although they still most likely require slight adjustment before they can be embedded into the new configurator platform. The calculation modules have internal linkings to each other and a few external databases. The external dependencies need to be stripped from the components prior they can be included into the new configurator in a reasonable manner. This operation can be performed asynchronously with the rest of the sales configurator development and it could be initiated instantaneously to speed up the entire configurator project. Even though the calculation modules are not intended to be rewritten in the near future, it should be noted that the Visual Basic 6 technology behind all of them is getting old and has already lost its technical support, which is why plans for their redesign ought to be made to ensure the future compatibility.

In addition to the regular project management aspects, the configurator project could benefit from other elements, such as project web pages, since one of the results that indirectly came out from the answers of business study was a fear among the salesmen that the future system would somehow be inferior when compared to the old one. This fear of systems turning into worse, could be addressed by establishing a project web page or a development diary that would explain the most recent development situation and the reasoning behind certain decisions so that the salesmen and interest groups

could have a sense of involvement during the project. The transparency in the project should also reduce the fear of change among the users and might act as an additional safety net to ensure that all the needed features are implemented.

8.2 Final thoughts

According to a proverb every victory has a taste that is bittersweet, which also held true in this thesis. The hardest part of this thesis was to leave any of the technologies out from the final selection, not only because many of the configurator vendors had put extensive effort for crafting an offer, but also because the different solutions varied so much from each other that finding evaluation criteria which would treat each contestant fairly was a difficult task and required quite a lot of adjusting. Retrospectively, the final evaluation criteria should have been given more emphasis during the earlier parts of the study, preferably even before interviewing any of the configurator contestants.

Quite interestingly the study that began as a technology roadmap and evaluation eventually turned up putting more weight to humane aspects of the software development against the cameralistic view of pure technology evaluation. The more the different vendors were interviewed and explained the needs of Konecranes', the more it became apparent that the best solution might not be the one that has the most elegant technological approach, instead the one that is most flexible to adjust to the constantly changing target situation. In an ideal project, the target situation would stay fixed and constant during the development project, however, in a large-scale software project as a sales configurator is, it appears to be fairly difficult a task to specify all the different needs so thoroughly that they will be understood correctly.

Looking back, the things that could have been done better during this study mainly revolve around the final selection criteria and its construction. Critically thinking, more attention should have been paid to the final selection criteria at the earlier stages of this project, although it might have been difficult due to the fact that advanced sales configurators are not standard off-the shelf solutions, and the different alternatives diverge so much from each other in terms of benefits and weaknesses that finding an objective evaluation criteria that is not affected by fundamental differences between the solution proposals is difficult. Defining key requirements before contacting any of the vendors turned out to be a good solution, as it allowed specifying the outlines for the project, and helped to leave out those vendors that could not have been able to provide required solution in the first place.

However, if more attention to the final evaluation criteria would have been paid before interviewing any of the vendors, more likely it would have made it possible to seek correct answers for related questions. Also, paying more attention to details, such as providing a calculation module and a simple product structure to each of the contestants for testing purposes would have helped a lot when estimating whether a solution proposal is able to meet the demands or not. Finding the optimal evaluation criteria, where all the different qualities would have had a quantitative and easily measurable value, would have probably been impossible or even unpractical, instead finding criteria that won't be affected when the target situation changes when the project definition changes could have led into more specific results.

On the other hand, objective metrics in the evaluation could have also been difficult to achieve due to the fact that each of the vendor contacts had a different approach to the subject, and each of the people representing the companies had a different technical and educational approach, some being software engineers and the other being sales persons. Much effort was paid during this thesis to get to the interviews with different people that represented both, technological and sales oriented specialists, of the vendor so that the both views could be thoroughly examined, which intuitively turned out to be a good solution since service commitment eventually turned out to weight as much as technological aspects in this study.

The academic part of this study clearly was a needed one, most notably for the reason that it offered a general understanding to a world of configurators and helped to set a goal for the features that could be expected from any decent configurator vendor. It particularly helped in the earlier stages to find a mutual understanding and terminology to communicate with the vendors, in order to find out which ones had a deeper understanding in complex configuration tasks, and which of the companies were offering merely an electronic price catalogue. Those vendors that had a deeper understanding towards the subject were not surprised about the questions relating constraint satisfaction based reasoning in the configuration engine, or other matters covered in the theoretical part of this study.

One of the most difficult parts of the thesis was to separate veritable configurator companies out of the ones that offered only simplified solutions for uncomplicated needs. The companies that made it to the final evaluation turned out to meet the most complex demands, which is also why eventually the hardest part of this thesis turned out to be to leave any of the contestants out. As Edsger W. Dijkstra once described: "Computer Science is no more about computers than astronomy is about telescopes."

REFERENCES

- [1] Roivainen, Tero. 2008. Sales Systems Manager. Konecranes Standard Lifting, Hämeenlinna, Finland. Guidance during the thesis.
- [2] Konecranes International Plc. 2008. Annual Report 2007.
- [3] Lohikoski, Niilo. 2007. Myynnin tukijärjestelmien kehittäminen liiketoimintaa tukevaksi kokonaisuudeksi. Master's thesis. Espoo. Helsinki University of Technology.
- [4] Konecranes International Plc. 2007. Corporate IT-days Presentation Material.
- [5] Sääksjärvi, Pekka. 2008. Markman 2000 project manager. Konecranes Standard Lifting, Hämeenlinna, Finland. Interviews during the thesis.
- [6] Sarinko, Kati. 1999. Massaräätälöinti, konfigurointi ja modulointi asiakaskohtaisesti muunneltavien tuotteiden hallinnassa. Master's thesis. Helsinki University of Technology.
- [7] Blecker, T. et al. 2007. Mass Customization Information Systems in Business. Hershey PA, IGI Publishing, Information Science Reference. 313 p. ISBN 978-1-59904-041-7.
- [8] Lamberg Rainer. 2003. Tuotannonohjaus. Stadia. [www-document] 1st September 2003 [referenced 13th March 2008]. Available at: <<http://cs.stadia.fi/~lamberg/ryhmat/SCIY4/tuotantomuodot.html> >
- [9] Kentta, Sami. 2008. Development Manager, CAD systems. Konecranes Corporation Plc, Hyvinkää, Finland. Interview in February 18th 2008.
- [10] Helo, P., Kyllonen, S., Jiao, R. 2007. Industrial Engineering and Engineering Management, 2007 IEEE International Conference on 2-4 Dec. 2007. Pages: 1302-1306. ISBN: 978-1-4244-1529-8.
- [11] Papinniemi, Jorma. 2006. Lecture Notes. Lappeenranta University of Technology, Tuotetiedonhallinta.
- [12] Redlein, A. 1999. Fulfilling customer's needs by the use of a variant configurator for dynamic product definition. Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on 18-21 Oct. 1999, Volume 1, Pages: 735-742. ISBN: 0-7803-5670-5.
- [13] Helin, Juha. 2008. IT manager, Erp systems, Hämeenlinna, Finland. Interview in March 21st 2008.
- [14] Sääksvuori, A. Immonen, A. 2002. Tuotetiedonhallinta - PDM. Talentum Media Oy, Helsinki. 201 p. ISBN: 951-762-796-3
- [15] Ojala, Jussi. 2008. Myyntijärjestelmien ja -prosessin kehittäminen globaalissa nosturi- ja nosturikomponenttiyrityksessä. Master's thesis. Tampere University of Technology.

- [16] UGS Corp. 2006. JT File Format Reference, Version 8.1, Revision B. [e-book] Available at: <http://www.jtopen.com/docs/JT_File_Format_Reference_Rev-B.pdf>
- [17] UK Auto Industry. 2008. VW and Audi to use Siemens PLM Software's Teamcenter product data management platform. Industry News [www-document]. 26th February 2008 [referenced 13th March 2008]. Available at: <http://www.autoindustry.co.uk/news/26-02-08_2>
- [18] Siemens PLM Software. 2008. Teamcenter Overview Brochure. [www-document, referenced 26th February 2008]. Available at: <http://www.plm.automation.siemens.com/en_gb/Images/teamcenter_overview_broch_W22_tcm642-3245.pdf>
- [19] Puustjärvi, Juha. 2007. Lecture Notes. Lappeenranta University of Technology, Web teknologiat.
- [20] Berners-Lee, Timothy. 1999. Weaving the Web: Origins and Future of the World Wide Web. Great Britain, Harper Collins Publishers, 1999. ISBN 0-7528-2090-7.
- [21] The Open Group. 2008. Service Oriented Architecture. [www-document, referenced 20th March]. Available at: <<http://www.opengroup.org/projects/soa/>>
- [22] Wahli, U. et al. 2006. Web Services Handbook for WebSphere Application Server 6.1. IBM Corporation. 789 p. ISBN: 0738494909
- [23] Bramham, J., MacCarthy, B. 2004. The demand driven chain. Mass Customisation Res. Centre, Nottingham Univ. Bus. Sch., UK., Manufacturing Engineer, Volume 83, Pages: 30-33. ISSN: 0956-9944
- [24] Yue Wang, Tseng, Mitchell M. 2007. An approach to improve the efficiency of configurators. Industrial Engineering and Engineering Management, 2007 IEEE International Conference on 2-4 Dec. 2007, Pages: 1332-1336. ISBN: 978-1-4244-1529-8
- [25] Engelmores, R., Feigenbaum, E. 1993. Knowledge based systems in Japan. Japanese Technology Evaluation Center. Loyola College, Maryland. [www-document] May 1993 [referenced 17th October 2008] Available at: <<http://www.wtec.org/loyola/kb/toc.htm>>
- [26] Mittal, S., Frayman, F. 1989. Towards a generic model of configuration tasks. Proceedings of International Joint Conference on Artificial Intelligence, California, Morgan Kaufmann, 1989, Pages: 1395-1401. ISSN: 1045-0823
- [27] Helander, M., Jiao, J. 2006. Development of an electronic configure-to-order platform for customized product development. Computers in Industry, Volume 57, Issue 3. Elsevier Science, 2006, Pages: 231-244. ISSN: 0166-3615
- [28] Felfernig, A., Friedrich, G., Jannach, D. 2001. Conceptual modeling for configuration of mass-customizable products. Artificial Intelligence in Engineering, Volume 15, Issue 2. Elsevier Science, 2001, Pages: 231-244. ISSN: 0954-1810

- [29] Haikala, I. Märijärvi, J. 2003. Ohjelmistotuotanto. Pieksämäki, Talentum Media Oy. 2003. 429 p. ISBN 952-14-0486-8
- [30] Ferguson, Scott. The Birth of Visual Basic. [www-document, referenced June 2008] Available at: <<http://www.forestmoon.com/BIRTHofVB/BIRTHofVB.html>>
- [31] TIOBE Software BV. 2007. TIOBE Programming Community Index for September 2007. [www-document] September 2007. [referenced June 2008] Available at: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>
- [32] Martin, James. 1991. Rapid Application Development. Macmillan Publishing Co, USA. 788 p. ISBN:0-02-376775-8
- [33] Product Family Life-Cycle Guidelines for Visual Basic 6.0. 2008. Microsoft Corporation. [www-document, referenced July 2008] Available at: <<http://msdn.microsoft.com/en-us/vbrun/ms788707.aspx>>
- [34] Kuitunen, Jarmo. 2008. Product Manager. Konecranes Standard Lifting, Hämeenlinna, Finland. Interviews in April 2008.
- [35] Vuorio, Juho. 2006. Konfigurointiprosessin kehittäminen tuotteistetulle nostovaunulle. Master's thesis. Helsinki March 30th 2006. Helsinki University of Technology.
- [36] The Compendium of Professional Selling (CoPS). 2005. United Professional Sales Association: Professional Practices. [www-document, referenced June 2008] Available at: <<http://www.upsa-intl.org/index-4-cops.htm>>
- [37] IEEE Standards Board. 1993. IEEE Standard 830, Guide to Software Requirements Specifications. Institute of Electrical and Electronics Engineers, Inc. December 2nd 1993. ISBN: 1-55937-395-4.
- [38] Salmi, Ilkka. 2008. Finnish Security Police - Annual Report. [www-document] Updated 2008. [referenced May 2008] Available at: <<http://www.supo.fi/poliisi/supo/home.nsf/pages/2EF2F56EFD9739A3C225715D0044B638?opendocument>>
- [39] Boehm, Barry. 1991. Software risk management: Principles and practices. IEEE Software, Volume 8, Issue 1, Pages: 32-41. ISSN: 0740-7459
- [40] Ropponen, J., Lyytinen, K. 2000. Components of software development risk: how to address them? A project manager survey. Software Engineering, IEEE Transactions. February 2000, Volume 26, Issue 2, Pages 98-112. ISSN: 0098-5589.
- [41] Ylikippari, Kai. 2008. Workstation Manager. Konecranes Corporation Plc. Interview in July 6th 2008.
- [42] Konecranes Business Portal Guidelines. 2007.
- [43] Wt, C++ Web Toolkit Introduction. [www-document] Updated September 12th 2008. [referenced July 2008] Available at: <<http://www.webtoolkit.eu>>

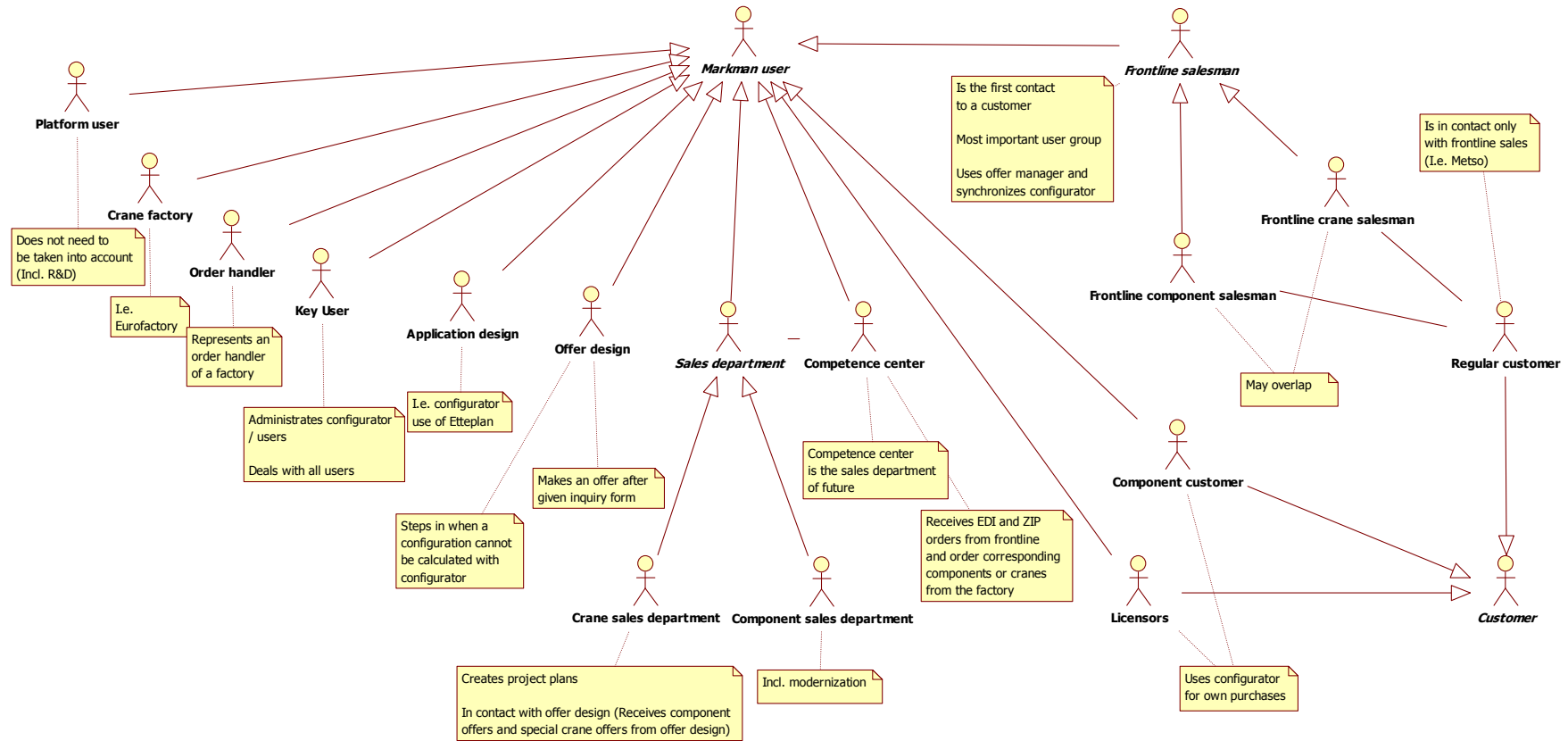
- [44] Microsoft Exchange Team. 2005. Outlook Web Access-A catalyst for web evolution. [www-document] Updated June 21st 2005 [referenced July 2008] Available at: <<http://msexchangeteam.com/archive/2005/06/21/406646.aspx>>
- [45] Google Corporation. 2008. Google Docs Tour. [www-document, referenced July 7th 2008] Available at: <<http://www.google.com/google-d-s/intl/en/tour1.html>>
- [46] Microsoft Corporation. 2005. ASP.NET AJAX Overview. [www-document] Updated 2007 [referenced July 2008] Available at: <<http://asp.net/ajax/documentation/live/Overview/default.aspx>>
- [47] Google Corporation. 2008. What Is Google App Engine?. [www-document, referenced July 2008] Available at: <<http://code.google.com/appengine/docs/whatisgoogleappengine.html>>
- [48] World Wide Web Consortium. 2008. Rich Web Client Activity Statement. [www-document] Updated April 2008. [referenced August 2008] Available at: <<http://www.w3.org/2006/rwc/Activity.html>>
- [49] Alvarez, Gene. 2007. MarketScope for Sales Configuration, 3Q07. Gartner Research. [www-document] Updated October 29th 2007. [referenced August 2008] Available at: <<http://mediaproducts.gartner.com/reprints/oracle/150194.html>>
- [50] Kaskela, Lauri. 2005. Tietotekniikkahankinnat - Hankintaprosessi. TIEKE Tietoyhteiskunnan kehittämiskeskus ry. [www-document] Updated August 5th 2005. [referenced August 2008] Available at: <http://www.tieke.fi/verkkokaveri/teemat/tietotekniikkahankinnat/tietotekniikan_hankinta/hankintaprosessi/--_text>
- [51] Tuominen, P. et al. 2008. Meeting with Wapice Oy. Hämeenlinna, March 20th 2008.
- [52] Wessman, N., Kukko, J. 2008. Meeting with NSD Consulting. Hämeenlinna, April 30th 2008.
- [53] Microsoft Corporation. 2008. Windows Presentation Foundation. [www-document, referenced May 2008] Available at: <<http://msdn.microsoft.com/en-us/netframework/aa663321.aspx>>
- [54] Hendrikson, Mike. 2008. State of the Computer Book Market. O'Reilly Publishing. [www-document] Updated February 20th 2008. [referenced May 2008] Available at: <<http://radar.oreilly.com/archives/2008/02/state-of-the-computer-book-mar-19.html>>
- [55] Orsvärn, K., Leitgeb, M. 2008. Meeting with Tacton Systems AB. Hämeenlinna, June 19th 2008.
- [56] Reiss, Markku. 2004. Modultek ja ruotsalaisikumppani tehostavat tuotehallintaa. Digitoday, May 10th 2004. [www-magazine] Available at: <<http://www.digitoday.fi/data/2004/05/10/modultek-ja-ruotsalaisikumppani-tehostavat-tuotehallintaa/20049973/66>>

- [57] Semenoja, M., Vehmanen J. 2008. Oracle response for sales configurator inquiry.
- [58] Ackermann, Philipp. 2008. Meeting with Perxpectix AG. Hämeenlinna, September 15th 2008.
- [59] Carlson, C., Fallows J. 2008. Meeting with Cincom representatives. Hämeenlinna, October 22nd 2008.
- [60] Ohr, S., Heiob, W. 2008. Presentation and meeting about Epos sales configurator. Künzelsau, September 9th 2008.
- [61] Martio, A., Anderson, A. 2008. Meeting with Variantum. Hämeenlinna, August 25th 2008.
- [62] Idola Fori. 2008. Presentation material of sales configurator.
- [63] Yang, J., Xu, D. 2002. On the evidential reasoning algorithm for multiattribute decision analysis under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 32, Issue 3, Pages 289-304. ISSN: 1083-4427
- [64] Nielsen, Jacob. 1995. *Alertbox: Current Issues in Web Usability*. [www-document] Updated November 3rd 2008. [referenced November 3rd 2008] Available at: < <http://www.useit.com/alertbox/> > ISSN: 1548-5552

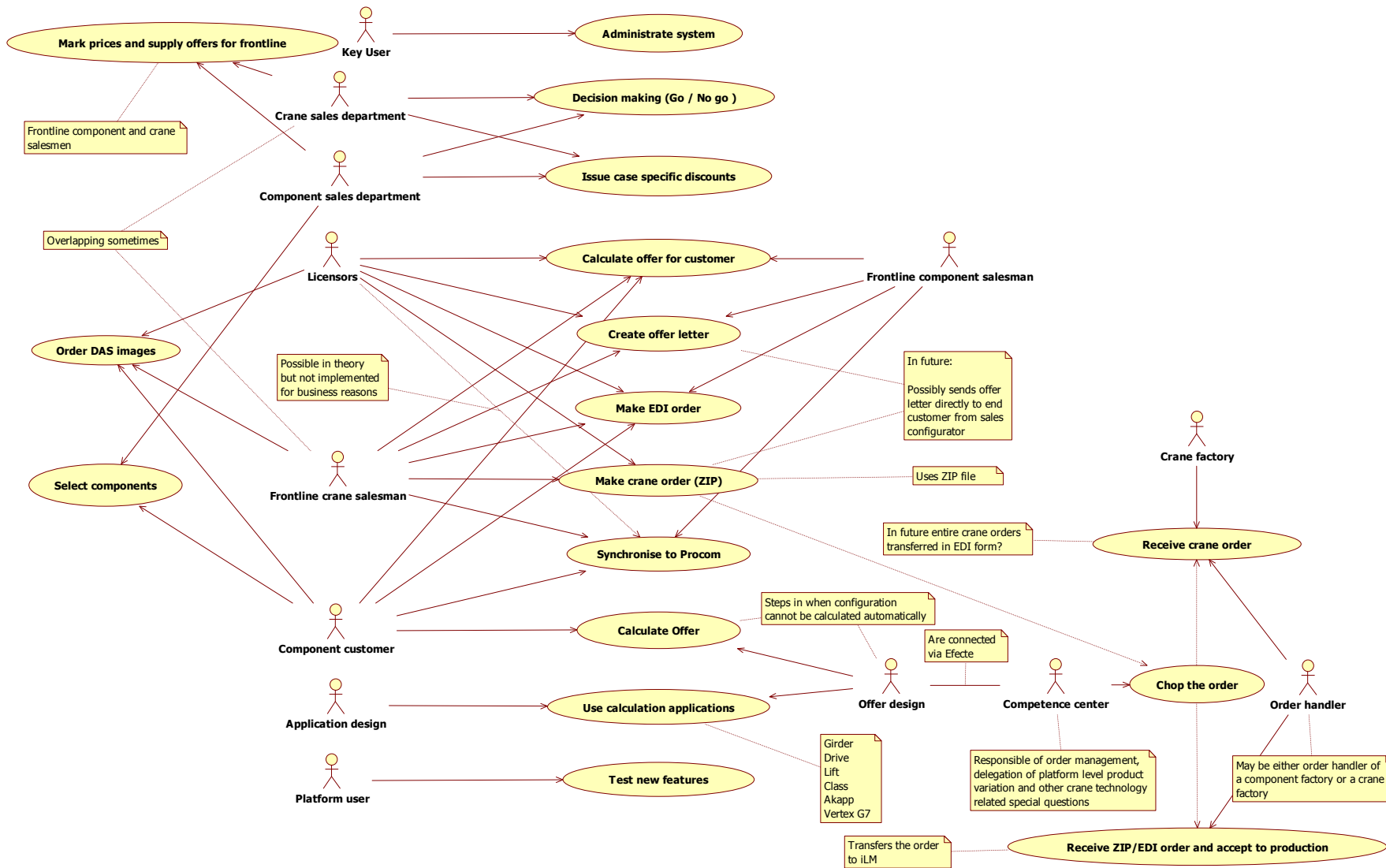
APPENDIX 1. Calculation modules of Markman 2000

Name	Responsibilities
AddCal.dll	Addition Calculation (Set electrical parameters)
AddFeaCosCls.dll	Additional Feature Cost Calculation (Sets prices for additional features)
BriPan_SM.dll	Bridge Panel Selection (For SM cranes)
BriPan_VS.dll	Bridge Panel Selection (For rest of cranes)
BufCal.dll	Crane Buffer Calculation (Selects correct buffer for crane according to environment constraints)
CraApr.dll	Crane Approach Calculation (Sets approach distance parameters)
CraDelTime.dll	Crane Delivery Time (Calculates time for crane delivery)
CraDrw.dll	Crane 2D Drawing (Steers drawing generation in CADman)
CraMea.dll	Crane Measures (Sets measure parameters for crane)
CraStdCal.dll	Standard Values for Crane Calculation (Sets the initial values for configuration)
CraVolCls.dll	Crane Electrical Component Voltages Selection according to crane voltage
DelCal.dll	Delivery Calculation (Sets delivery pricing parameters)
DriAda.dll	Bridge Drive Selection (Sets initial values for drive calculation sub-module)
DsnCal.dll	Design Calculation (Costs for electric and mechanical calculation)
EleParSet.dll	Sets Electrical Parameters (Location of electrical components for trolley)
EndSel.dll	Selects End Carriages and joint type used in main girder
FinCod.dll	Converts technical values into branded ones
GirCal_new.dll	Main Girder Calculation Module (Designs main girder)
HoiCal.dll	Hoist Calculation Module (Calculates dimensions for hoists and related parameters)
HoiCal_QQ.dll	Same as above (for specific product family)
HoiCal_SM.dll	Same as above (for specific product family)
HoiCal_VS.dll	Same as above (for specific product family)
HoiCal_XC.dll	Same as above (for specific product family)
MM2000DriSel.dll	Drive Calculation Sub-Module (Selects optimised drive machinery)
PrdCraCal.dll	Crane Manufacturing Parameters
PrdGirCal.dll	Girder Manufacturing Parameters
PwrSupBri.dll	Crane Power Supply Calculation (Sets parameters for crane intake currency)
PwrSupDuctorCls.dll	Same as above (for specific product family)
PwrSupTro_VS.dll	Hoist Power Supply Calculation
TroAda.dll	Trolley Drive Selection (Sets initial values for drive calculation sub-module)
WheLoa.dll	Calculates wheel loads for crane end carriages
PusBtnNew.dll	Selects Controller type

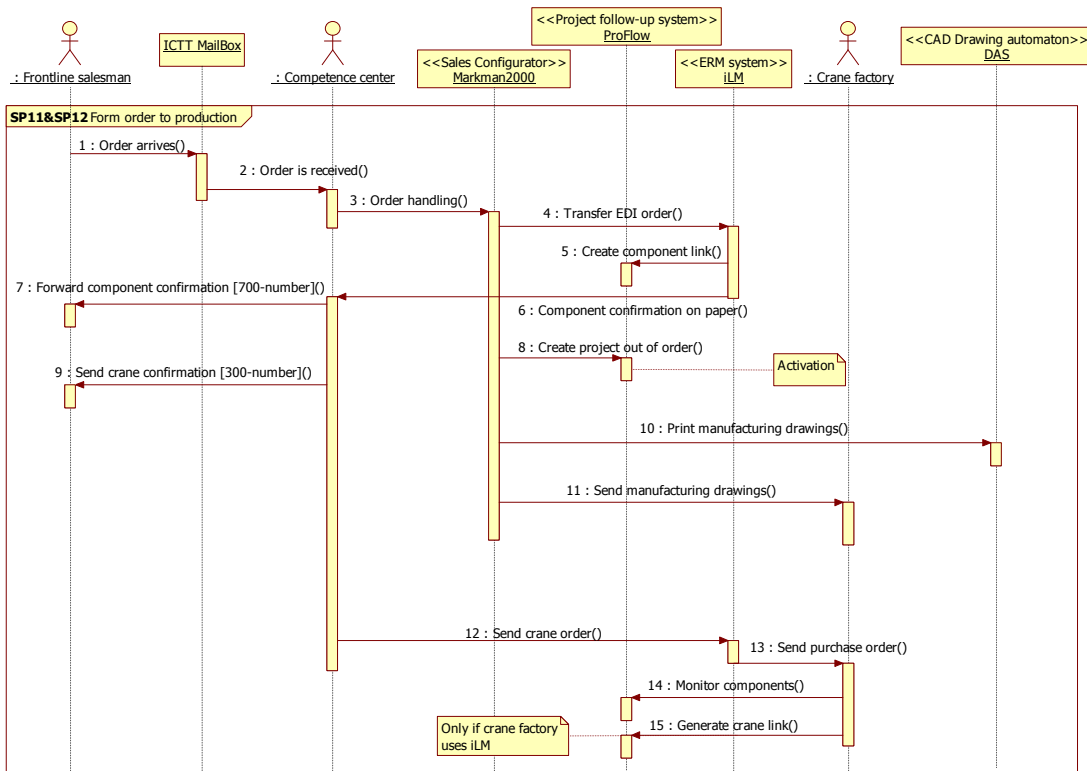
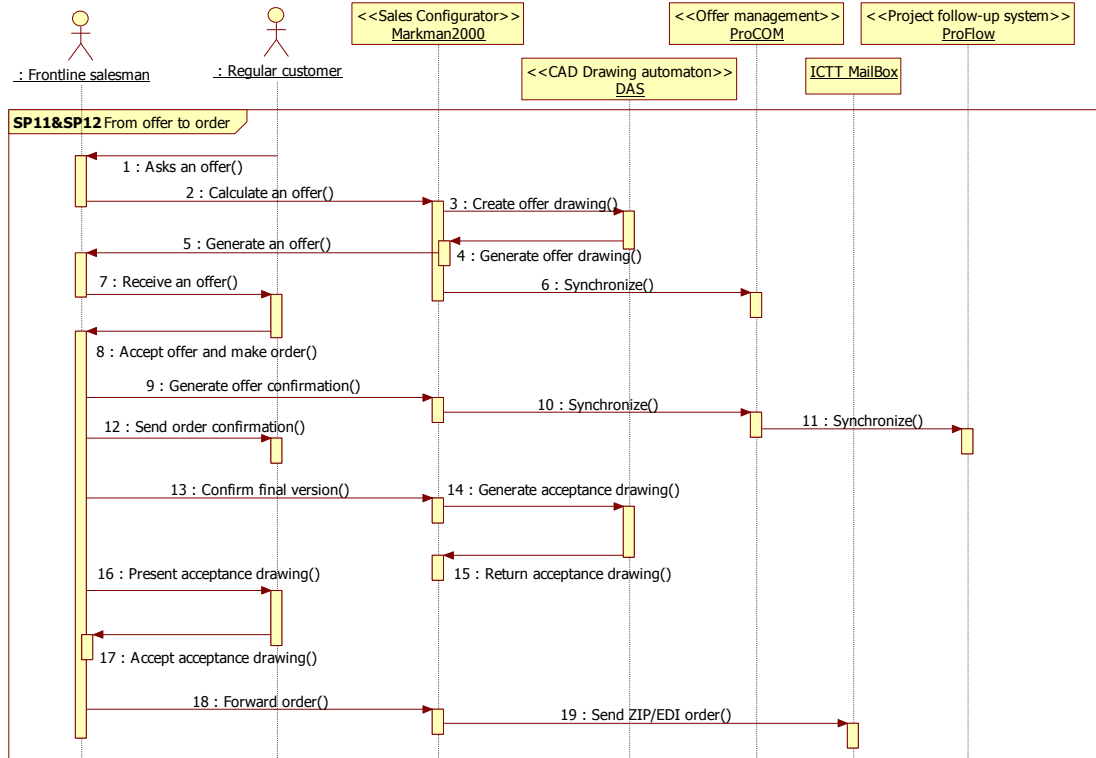
APPENDIX 2. User group context of MM2000



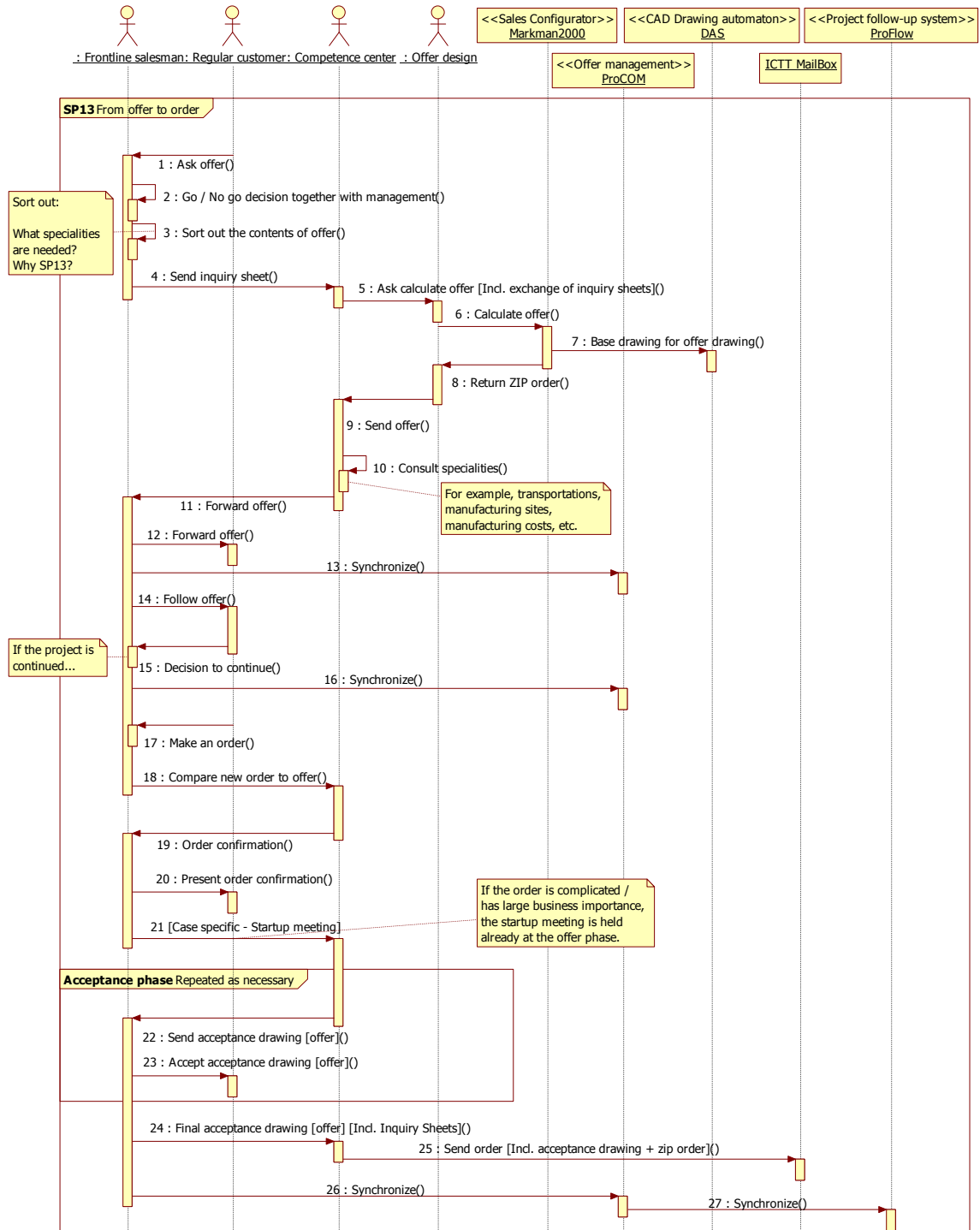
APPENDIX 3. Use cases of MM2000



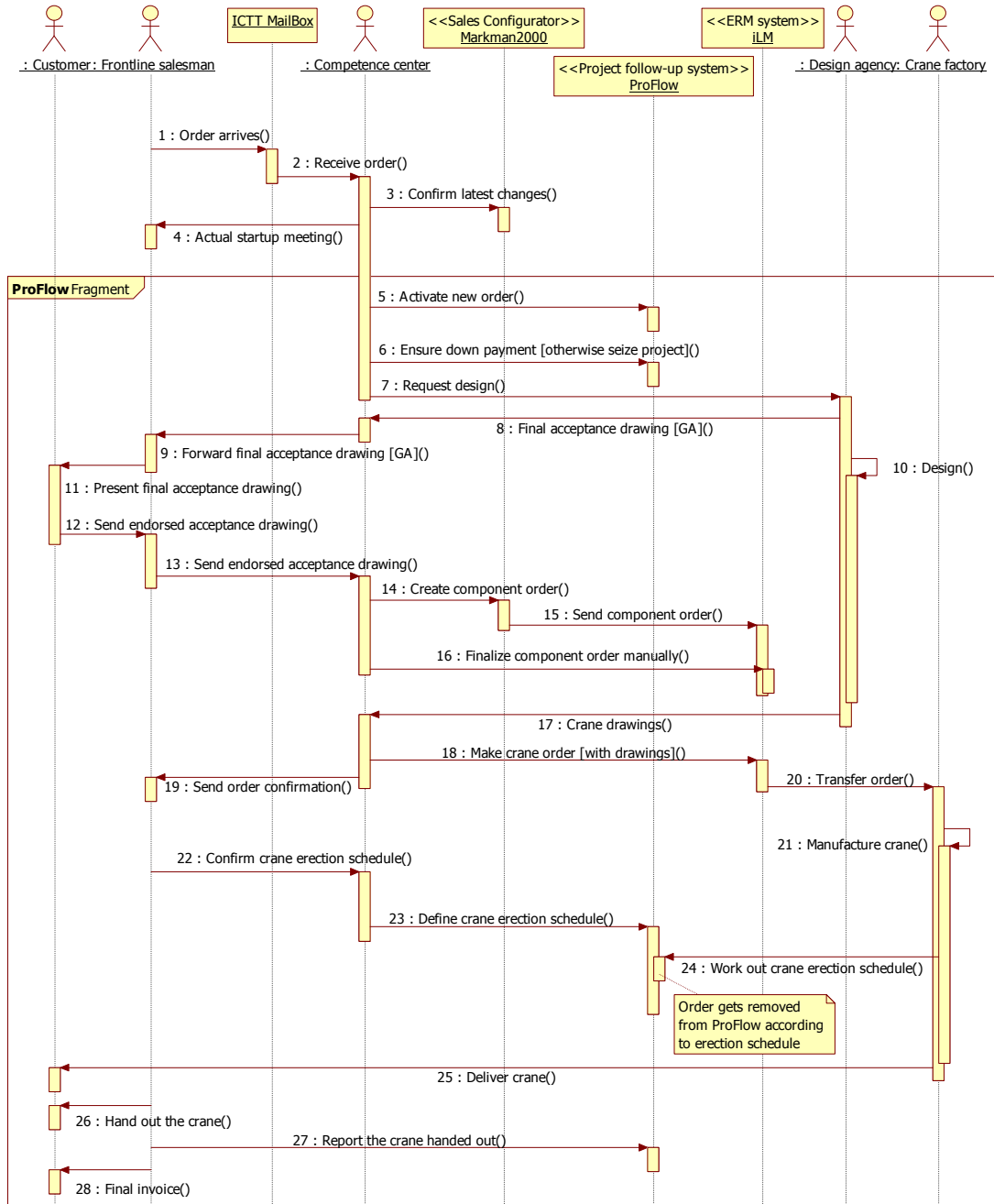
APPENDIX 4. SP11 & SP12 offer process flows



APPENDIX 5. SP13 Order to offer process flow



APPENDIX 6. SP13 Offer to delivery process flow



APPENDIX 7. Selection criteria with different weighting factors

Id	Criteria	Meaning of criteria	Weight
1	Comparison to theory		
1.1	Separating product knowledge from information technology	One of the fundamental tasks is to separate product knowledge from software knowledge.	2.5
1.2	Architectural comparability to theory	Similarity to presented architectures and features, such as GPF master model.	1.0
1.3	Constraint based configuration engine	What kind of rule engine does the configurator have?	1.0
2	Characteristics of solution		
2.1	How much of the work is outsourced	Most of the software knowledge should be outsourced, leaving only crane related specifics and application supervision to Konecranes.	2.0
2.2	Meeting all the key requirements	Different vendors meet the primary requirements in different levels. Measures how punctually primary requirements are interpreted.	3.0
2.3	Portability	Is the configurator portable to several different platforms?	0.5
2.4	Usability	Application usability according to common usability metrics. (Jacob Nielsen – Alert box)	2.5
2.5	Adaptability to changing IT systems	Can the configurator be rapidly changed to work with future CRM and CAD systems?	2.5
2.6	How are the products modelled?	Are there specific strengths in the product modelling techniques?	2.0
2.7	How are the optimisations handled	Does the configurator allow the product to be optimized according to various different attributes? (price, weight, delivery time)	1.5
3	Technology		
3.1	Compatibility with existing enterprise systems	How well can we expect the configurator to work with ERP, PDM, and ProCom systems?	2.5
3.2	Programming technique	Are the programming techniques likely to suit the purposes? Is the configurator able to function in restricting desktop environments? Does the application need frequent updates, for example, when user interface changes?	2.0
3.3	Fitting to desktop environment	Does the configurator work with limited user privileges and HP controlled desktop environment.	2.0
3.4	How much work is required to fit the configurator into other enterprise systems	Does the configurator fit easily into existing systems, or is special crafting or new adaptor systems needed? Can the product information be read from TeamCenter PLM?	2.5

APPENDIX 7. (Continues)

3.5	Openness of the system	Does the system provide any means of accessing it externally? Does the system have any open interfaces that allow controlling the application from other applications?	3.0
3.6	Speed of functionality	The system ought to be faster than the previous version. The speed includes the responsiveness of user interface as well as the actual configuration calculation.	2.5
3.7	Standards compliance	The application should utilise standardized methods in its design, data transfer, information storage, etc.	2.0
3.8	Will tie hands to a single architecture	The configurator solution should not limit the future possibilities regarding hardware or software.	2.2
3.9	Will tie hands to one system supplier	The configurator solutions should not limit the future vendors of other systems.	2.5
3.10	Kludging	Does the configurator solution have parts that seemingly do not fit the demands but have been circulated somehow?	1.0
4	Business goals		
4.1	Fit IT-strategy	The configurator should fit the general IT-strategy of the corporation, which involve uniform corporate image, easiness and transparency of usage, for example.	2.5
4.2	Fit non key requirements of business study	The business study included some valid requirements that did not make it to the key requirements but bring extra value.	2.0
4.3	Time to market	How long does the implementation phase take?	2.5
4.4	Update speed	How fast can products added to the system? How fast can the system react to changes?	2.5
5	Support and services		
5.1	After implementation support	What kind of service do we get after the implementation is complete? Is the vendor able and willing to provide support?	3.0
5.2	Documentation and training	What kind of documentation does the vendor provide? User's manual? Administration manual? Product maintenance manual? Does the configurator vendor provide training for product modellers or administrators?	2.5
5.3	Vendor proximity / commitment	The vendor should be able to provide service quickly in case of emergencies.	3.0
5.4	Help with merging calculation modules	The calculation modules need to be separated from the old configurator and building their new interfaces might require help.	2.5
5.5	Help with product models?	How much help is needed? Will we get any help? How much it costs?	2.0

APPENDIX 7. (Continues)

6	Costs		
6.1	Total cost to first finished product	How much do we expect to pay before the first finished version? (A working version with all the current products included)	2.0
6.2	After implementation costs	How much do we expect to have to pay for the maintenance after the configurator is deployed?	2.0
6.3	Payback time & Return of invest (ROI)	How long will it take before the configurator development begins to pay back? (Time to first usable version that brings additional value) How long will it take before the configurator is expected to pay back the investment?	4.0
7	Customization capabilities		
7.1	Customizable user interface?	Can different brands make their own layouts and documents to the application?	2.5
7.2	How much customization is needed?	Does the product need a lot of customization before it can be used for its purpose?	2.5
8	Vendor strength		
8.1	Industrial knowledge	Does the vendor have any previous experience dealing with similar problems?	2.2
8.2	Financial strength	The vendor should be financially on solid background to prevent it going out of the business.	2.5
8.3	Customer references	Does the vendor have any feasible customer references? If yes, what would they say about the product? (benchmarking)	2.0
8.4	Vendor resources	Do they have enough qualified personnel?	3.0
9	Vendor vision		
9.1	Vendor policy & philosophy	Does the general philosophy of the vendor fit to general philosophy of Konecranes? (future prospects, commitment)	1.0
9.2	Vendor R&D efforts towards renewing the configurator	Is the vendor actively developing the product and determined to improve it in future?	2.5
9.3	Vendor versatility	Flexibility & responsiveness	3.0
10	Risks		
10.1	Uncertainty to get finished product	Does vendor perform any project risk mitigation actions (Auditing, Certificates)	2.0
10.2	Information security & piracy	Trustworthiness of vendor.	3.0
10.3	Ability to perform in unexpected situations	Is the vendor able to handle unexpected situations in case the project encounters such? Enough qualified personnel and strong service commitment?	2.5
11	Other features		
11.1	Consulting in other software needs	Is the vendor willing to assist in other software problems, such as additional tools?	2.0

APPENDIX 8. Grading of sales configurator vendors

G = grade given to vendor

W = grade multiplied by weight factor (final grade)

Criterion	Weight	Vendor 1		Vendor 2		Vendor 3		Vendor 4		Vendor 5		Vendor 6		Vendor 7		Vendor 8		Vendor 9	
		G	W	G	W	G	W	G	W	G	W	G	W	G	W	G	W	G	W
1.1	2,5	4	10,0	2	5,0	5	12,5	5	12,5	5	12,5	4	10,0	4	10,0	2	5,0	5	12,5
1.2	1,0	4	4,0	1	1,0	5	5,0	5	5,0	4	4,0	5	5,0	5	5,0	1	1,0	5	5,0
1.3	1,0	3	3,0	0	0,0	5	5,0	5	5,0	3	3,0	5	5,0	4	4,0	0	0,0	2	2,0
2.1	2,0	2	4,0	3	6,0	4	8,0	2	4,0	2	4,0	4	8,0	1	2,0	5	10,0	4	8,0
2.2	3,0	5	15,0	4	12,0	4	12,0	1	3,0	4	12,0	5	15,0	3	9,0	1	3,0	3	9,0
2.3	0,5	4	2,0	2	1,0	4	2,0	5	2,5	5	2,5	5	2,5	2	1,0	3	1,5	5	2,5
2.4	2,5	2	5,0	4	10,0	5	12,5	3	7,5	4	10,0	3	7,5	3	7,5	3	7,5	3	7,5
2.5	2,5	3	7,5	4	10,0	4	10,0	2	5,0	4	10,0	4	10,0	3	7,5	3	7,5	4	10,0
2.6	2,0	5	10,0	2	4,0	3	6,0	4	8,0	2	4,0	5	10,0	3	6,0	0	0,0	2	4,0
2.7	1,5	3	4,5	4	6,0	3	4,5	3	4,5	3	4,5	4	6,0	2	3,0	3	4,5	2	3,0
3.1	2,5	3	7,5	5	12,5	4	10,0	2	5,0	3	7,5	3	7,5	3	7,5	1	2,5	4	10,0
3.2	2,0	3	6,0	3	6,0	3	6,0	3	6,0	5	10,0	3	6,0	4	8,0	3	6,0	3	6,0
3.3	2,0	3	6,0	5	10,0	4	8,0	2	4,0	5	10,0	3	6,0	3	6,0	3	6,0	3	6,0
3.4	2,5	3	7,5	5	12,5	4	10,0	1	2,5	3	7,5	3	7,5	3	7,5	2	5,0	3	7,5
3.5	3,0	2	6,0	3	9,0	5	15,0	3	9,0	4	12,0	3	9,0	4	12,0	1	3,0	3	9,0
3.6	2,5	3	7,5	5	12,5	3	7,5	4	10,0	2	5,0	3	7,5	4	10,0	3	7,5	2	5,0
3.7	2,0	4	8,0	3	6,0	3	6,0	3	6,0	5	10,0	4	8,0	3	6,0	2	4,0	4	8,0
3.8	2,2	2	4,4	3	6,6	4	8,8	4	8,8	2	4,4	4	8,8	3	6,6	2	4,4	3	6,6
3.9	2,5	2	5,0	3	7,5	5	12,5	3	7,5	3	7,5	4	10,0	3	7,5	2	5,0	3	7,5
3.10	1,0	5	5,0	5	5,0	3	3,0	2	2,0	2	2,0	4	4,0	3	3,0	2	2,0	1	1,0
4.1	2,5	3	7,5	3	7,5	3	7,5	1	2,5	5	12,5	3	7,5	2	5,0	1	2,5	4	10,0
4.2	2,0	2	4,0	3	6,0	4	8,0	1	2,0	3	6,0	3	6,0	2	4,0	2	4,0	3	6,0
4.3	2,5	3	7,5	3	7,5	3	7,5	3	7,5	3	7,5	4	10,0	3	7,5	3	7,5	3	7,5
4.4	2,5	2	5,0	5	12,5	3	7,5	2	5,0	3	7,5	3	7,5	1	2,5	3	7,5	3	7,5

APPENDIX 8. (Continues)

G = grade given to vendor

W = grade multiplied by weight factor (final grade)

5.1	3,0	2	6,0	3	9,0	3	9,0	0	0,0	3	9,0	3	9,0	2	6,0	2	6,0	4	12,0
5.2	2,5	4	10,0	1	2,5	4	10,0	3	7,5	4	10,0	5	12,5	4	10,0	1	2,5	3	7,5
5.3	3,0	2	6,0	4	12,0	4	12,0	0	0,0	2	6,0	3	9,0	3	9,0	2	6,0	5	15,0
5.4	2,5	1	2,5	5	12,5	4	10,0	2	5,0	1	2,5	3	7,5	4	10,0	2	5,0	5	12,5
5.5	2,0	5	10,0	3	6,0	4	8,0	2	4,0	2	4,0	3	6,0	3	6,0	3	6,0	3	6,0
6.1	2,0	3	6,0	4	8,0	3	6,0	3	6,0	3	6,0	3	6,0	4	8,0	5	10,0	4	8,0
6.2	2,0	2	4,0	2	4,0	3	6,0	3	6,0	2	4,0	3	6,0	3	6,0	5	10,0	4	8,0
6.3	4,0	3	12,0	4	16,0	3	12,0	2	8,0	3	12,0	4	16,0	2	8,0	3	12,0	3	12,0
7.1	2,5	2	5,0	1	2,5	3	7,5	2	5,0	4	10,0	5	12,5	2	5,0	0	0,0	3	7,5
7.2	2,5	3	7,5	5	12,5	3	7,5	0	0,0	2	5,0	4	10,0	3	7,5	2	5,0	3	7,5
8.1	2,2	5	11,0	4	8,8	4	8,8	4	8,8	5	11,0	4	8,8	4	8,8	3	6,6	4	8,8
8.2	2,5	4	10,0	4	10,0	4	10,0	2	5,0	5	12,5	4	10,0	3	7,5	1	2,5	5	12,5
8.3	2,0	5	10,0	5	10,0	4	8,0	1	2,0	5	10,0	5	10,0	4	8,0	3	6,0	4	8,0
8.4	3,0	4	12,0	3	9,0	3	9,0	0	0,0	5	15,0	4	12,0	2	6,0	3	9,0	5	15,0
9.1	1,0	3	3,0	2	2,0	5	5,0	2	2,0	4	4,0	4	4,0	3	3,0	0	0,0	4	4,0
9.2	2,5	5	12,5	3	7,5	4	10,0	5	12,5	5	12,5	4	10,0	4	10,0	1	2,5	4	10,0
9.3	3,0	2	6,0	3	9,0	3	9,0	1	3,0	1	3,0	3	9,0	3	9,0	1	3,0	5	15,0
10.1	2,0	3	6,0	4	8,0	4	8,0	2	4,0	4	8,0	4	8,0	3	6,0	1	2,0	4	8,0
10.2	3,0	5	15,0	5	15,0	5	15,0	4	12,0	5	15,0	3	9,0	4	12,0	1	3,0	5	15,0
10.3	2,5	2	5,0	4	10,0	5	12,5	1	2,5	3	7,5	3	7,5	2	5,0	1	2,5	5	12,5
11.1	2,0	0	0,0	4	8,0	2	4,0	0	0,0	0	0,0	2	4,0	0	0,0	0	0,0	5	10,0
Total	101,9	140	311,4	153	358,4	170	382,1	108	228,1	152	342,9	167	371,1	133	298,9	91	206,5	164	375,9