SALES AND PRODUCT CONFIGURATOR
– A COMPARATIVE STUDY

Patrik Gustafsson and Jonas Lindholm

Programme for Computer Engineering 180 credits

Örebro, Sweden, Spring 2010

Examiner: Thomas Padron-McCarthy

SÄLJ- OCH PRODUKTKONFIGURATOR - EN JÄMFÖRANDE STUDIE
Abstract

Today companies in the manufacturing industries need to do everything possible to stay ahead in their market, in order to make a profit. This means that they have to slim the organization and every employee has to be more efficient. In other words the companies have to accomplish more with less people.

One way to rationalize the sales process is by using a sales and product configurator which is a software tool for handling products and its various variants by using rules and constraints in the configuration process.

In this paper a background on why configurators are needed is presented, along with some theory behind the configurator software.

Logica – a business and technology service company – gave us the task to make a comparative study of OTS (of the shelf) sales and product configurators, due to the increasing interest from the market.

Originally the study should have been done on configuration software but this could not be arranged. Instead information was gathered from the Internet, literature, scientific journals, and proceedings from international conferences in order to make a comparative study. Interviews with persons from two Swedish companies, Tacton Systems AB and In-process AB, which develop and sell configurators, were also conducted to get this information.

The report accounts for the result of the answers and their value.
Sammanfattning


Ett sätt att rationalisera säljprocessen är att använda en konfigurator som är ett mjukvaruverktyg för att hantera produkter och dess olika varianter med hjälp av regler och begränsningar i konfigurationsprocessen.

I denna uppsats presenteras en bakgrund till varför man behöver konfiguratorer, samt en del teori om konfiguratororns programvara.

Logica – ett IT-tjänsteföretag – har gett oss i uppdrag att genomföra en jämförelsestudie av sälj- och produktkonfiguratorer, på grund av ökat intresse från marknaden.

Ursprungligen skulle undersökningen ha gjorts på konfiguratorers programvara men detta kunde inte ordnas. För att få information för att besvara dessa frågor är information insamlad från Internet, litteratur, vetenskapliga tidskrifter och artiklar från internationella konferenser. Intervjuer med personer från två svenska företag, Tacton Systems AB och Inprocess AB, som utvecklar och säljer konfiguratorer, har utförts för att få denna information.

Rapporten kommer att redogöra för resultatet och dess värde.
Preface

The persons we would like to thank is

Ph.D. Klas Orsvärn, Tacton System AB
Martin Nilsson, In-process Sweden AB
Kjell Green, In-process Sweden AB

The interviews and discussions on configurators gave us invaluable information that would otherwise be difficult to get hold of.

And we would also like to thank our supervisors for their support

Kjell Mårdensjö, Örebro University
Erik Kristianssen, Logica AB, Örebro

For all the help with proofreading we would like to thank

Linda Johansson

Örebro 2nd of June 2010

Patrik Gustafsson and Jonas Lindholm
4.2.3.8 How does the modeling of the object that will be configured look like? .............................................. 28  
4.2.3.9 At what level of abstraction is the modeling? .................................................................................. 29  
4.2.3.10 What kind of support can you get from the software provider? ....................................................... 29  
4.2.3.11 What are the licenses costs? ........................................................................................................... 30  
4.2.3.12 Some other questions ................................................................................................................. 30  
4.3 COMBINUM CONFIGURATOR ............................................................................................................... 31  
4.3.1 The company In-process ..................................................................................................................... 31  
4.3.2 Different parts of the COMBINUM configurator system ..................................................................... 31  
4.3.3 The interview ....................................................................................................................................... 33  
4.3.3.1 What kind of limitations does a configurator have? ......................................................................... 33  
4.3.3.2 What interfaces are there against other systems and how complex and flexible are they? ................. 34  
4.3.3.3 How is the data handled? Interfacing with other systems? ................................................................. 36  
4.3.3.4 Is it true that high flexibility means high complexity? ..................................................................... 36  
4.3.3.5 How complex and flexible is the user interface? ............................................................................. 36  
4.3.3.6 How user-friendly is the configurator? ............................................................................................. 36  
4.3.3.7 Are there different types of complexity as the logic, user interface and/or API? ................................. 37  
4.3.3.8 How does the modeling of the object that will be configured look like? ........................................... 37  
4.3.3.9 At what level of abstraction is the modeling? .................................................................................. 37  
4.3.3.10 What kind of support can you get from the software provider? ..................................................... 38  
4.3.3.11 What are the licenses costs? .......................................................................................................... 38  
4.3.3.12 Some other questions ............................................................................................................... 38  
4.4 A SUMMARY OF THE INFORMATION FROM INTERVIEWS .................................................................... 40  
4.4.1 Q1. Which configurators are there on the market? ........................................................................... 40  
4.4.2 Q2. What kind of limitations does a configurator have? ................................................................... 40  
4.4.3 Q3. What interfaces are there against other systems and how complex and flexible are they? .......... 41  
4.4.4 Q4. Is it true that high flexibility means high complexity? ................................................................. 41  
4.4.5 Q5. How complex and flexible is the user interface? ....................................................................... 41  
4.4.6 Q6. How user-friendly is the configurator? ....................................................................................... 41  
4.4.7 Q7. Are there different types of complexity as the logic, user interface and/or API? ......................... 41  
4.4.8 Q8. How does the modeling of the object that will be configured look like? ................................... 42  
4.4.9 Q9. On which abstraction level is the modeling? ............................................................................. 42  
4.4.10 Q10. What kind of support can you get from the software vendor? .................................................. 42  
4.4.11 Q11. What are the licenses fees? ....................................................................................................... 42  
4.4.12 Summary of the information from the interviews in a table ............................................................. 43  
5 CONCLUSIONS ........................................................................................................................................ 44  
5.1 INTRODUCTION ................................................................................................................................. 44  
5.2 DID WE REACH OUR GOALS WITH THE STUDY? ........................................................................... 44  
5.3 WHAT CAN YOU DO WITH THE RESULT OF THE REPORT? ............................................................. 45  
5.4 HOW CAN THE STUDY GO FORWARD FROM THIS POINT? ............................................................. 45  
REFERENCES ............................................................................................................................................. 46  

APPENDIX

APPENDIX A Interview questions Tacton configurator.

APPENDIX B Interview questions COMBINUM configurator.

APPENDIX C Tacton configurator, an application example.

APPENDIX D COMBINUM configurator, an application example.
1 Introduction

1.1 Background

Today companies in the manufacturing industries need to do everything possible to stay ahead in their market, in order to make a profit. This means that they have to slim the organization and every employee has to be more efficient. In other words the companies have to accomplish more with less people.

When dealing with complex products that have many different variants in the sales process, it can take days or even weeks to make the calculations and get the information they need for preparing an offer.

The many different parts can be combined with different variants of products depending how the parts will be put together, meaning that a product will be configured to consist of different parts. The configuration process consists of choosing which parts the product will consist of and how they will be combined into the finished product.

One way to rationalize the sales process is by using a configurator which is a software-tool for handling products and different variants, by using rules and constraints in the configuration process. When using this tool in the right way the benefits can be the following according to Tiihonen [1]:

- It cuts down the number of errors in the configuration process.
- Reduces the lead-times in the sales-delivery process.
- Efficiently distribute up-to-date product knowledge to the people in the configuration process.
- Increase the volume of quotations without increasing the number of sales persons and engineers.

The term sales-delivery-process is used to describe all the phases; to propose, sell, order, manufacture, and finally deliver an individual product to the customer [1].

The IT systems that are used in companies today have moved from being support systems to being the core systems for the organizations. The software has matured from being software systems developed by consultant firms, to software that you can buy “off the shelf” (OTS), which you can adapt to the need of the organization.

The interests for OTS Product Configurators have increased in recent years and the companies see a future where such tools can be used to increase the sale.

One of the purposes with the report is to make a comparative study of OTS sales and product configurators.
1.2 Problem Discussion

Due to increasing interest for OTS “of the shelf” sales and product configurators in the market, the business and technology service company Logica has given us the task to find out the advantages and disadvantages of different types of Configurator systems and their complexity and flexibility, and also to make a deeper study of the sales and product configurator Tacton.

To get the information Logica are interested in, the following questions need answers:

Q1. Which configurators are there on the market?
Q2. What limitations does a configurator have?
Q3. What interface is there against other systems regarding complexity and flexibility?
Q4. Is it true that high flexibility means high complexity?
Q5. How flexible and complex is the user interface?
Q6. How user friendly is the configurator?
Q7. Are there different types of complexity, like the logic behind the user interface and/or API?
Q8. How is the modeling represented of the object that will be configurated?
Q9. On which abstraction level is the modeling?
Q10. What kind of support can you get from the software vendor?
Q11. What are the license fees?

The main purpose of our work has been to get answers to these questions, which will be presented in this report and also an insight into the area of sales and product configurators.

1.3 Research Methodology and Limitations

Information is gathered from the Internet, literature, scientific journals, and proceedings from international conferences.

In order to make a deep analysis of the Tacton configurator, that are developed by Tacton Systems AB, we were supposed to use Tacton configurator software provided by Logica. The purpose was to investigate the abilities of the configurator and - if we had sufficient time - make an example implementation.

Due to the fact that we were not able to get hold of a license on the Tacton configurator, we had to get the information we needed in another way.

The solution was to interview Dr. Klas Orsvärm, head of Research & Development at the Swedish company Tacton System AB that develop and sales the Tacton Configurator. We also interviewed Martin Nilsson and Kjell Green from In-process, another Swedish company that develops and sells the Combinum Configurator. Both companies were asked questions based on the information stated in section 1.2 Problem Discussion.
1.4 Source criticism

Almost all the material we have gathered is information from developers and sellers of the configurators. This makes it hard to get the disadvantages of the configurators due to the fact that they are trying to promote their products. They are also reluctant to show the faults with their products.

Since we were not able to fully test a configurator system ourselves, it is hard for us to verify all the claims from the developers of the Tacton and Combinum configurators.

1.5 Connection to our previous studies

We have benefited greatly from our previous studies in various subjects, mostly in the area of Artificial Intelligence but also other computer studies. This helped us to understand the information we gathered about configurators. It has been a very interesting project to see how different solutions can solve the same problem.
2 Why do we need configurators?

To see the advantages with Configurators one must understand what kind of problems they are supposed to solve. Here we will describe some background on why the configurator is needed.

2.1 Mass Customization

The last decades of competition in the manufacturing industry has made it hard for industries in non-low salary countries to make large profits with large volumes of mass-produced standard products. The demand from customers, for individualized products with short delivery time, is also increasing.

To satisfy the customer’s demand for individualized products, one-of-a-kind products take too much time to construct and produce, which makes the cost too high.

The solution is to use configurable products. These products have a large number of variants which can serve different customer needs by the possibility of customer-specific adaptation of the product. The goal is to combine the benefits of the mass-produced and the one-of-a-kind-products in a process sometimes called mass-customization [1].

According to Tiihonen[1] the Major trends in European industry are

- Diminishing lifetimes of products
- Increasing complexity and number of variants of products
- Need for shorter lead-times in the sales delivery process
- Increasing pressure from the customers and competitors to take individual customer requirements into account when specifying a customer-specific product instance
2.2 Configurable products

A configurable product is a type of a product that consists of a large number of variants and serves the specific needs of the individual customer by allowing customer-specific adaptation.

The products consist of different components. A component is a meaningful part of the product. The component itself can consist of one or more physical parts, see figure 1.

A pre-designed component is a re-usable specification which can be used in many different configurations of the products. Depending on what view you have a product could also be a component to another product. As in figure 1 a wheel can be a product A but also a component A in the product B a complete car.

According to Tiihonen in “Product Configurators -Information System Support for Configurable Products” [1] these are the basic characteristics for configurable products:

- Each delivered product instance is tailored to the individual needs of an individual customer.
- The product has been pre-designed to meet a given range of different customer requirements
- Each product instance is specified as a combination of pre-designed components or modules. Thus there is no need to design new components as a part of the sales-delivery process.
- The product has a pre-designed general structure or architecture or a set of these.
- There is no need to do creative or innovative design as a part of the sales-delivery process. Rather, the specification of a product instance can be done in a routine manner.

![Diagram](image)

**Figure 1.** Component A that consists of a Part A (Rim) and Part B (Tyre), can be a product A (wheel), or be included as a component in the Product B (complete car).
2.3 Product configuration process

Product configuration as an activity can be regarded as satisfying a customer’s demand by creating a product, which is composed by a number of pre-developed components.

Product configuration process is to convert customer needs into a product specification, quotation, requirements definition and manufacturing information.

2.4 Configurators as information system

A product configurator, or configurator for short, is an information system for managing products and their variants, and for doing customer-specific adaptation of the product. These systems will be described in more detail in the following chapter.
3 Configurators

3.1 History

Configurators as information systems have been used as an aid in the sales-delivery process since the beginning of the 1980’s. They stem from artificial intelligence technology and are considered to be one of its most successful applications [1].

The system that is often considered as the very first successful configuration system is Digital’s landmark R1/XCON system [14][19]. The R1/XCON was a production-rule based system written in OP5 by John P. McDermott of CMU in 1978 to assist in the ordering of DEC's VAX computer systems by automatically selecting the computer system components based on the customer requirements. Other organizations saw the potential and developed their own configuration systems. The programming paradigm that was used was known as production rules. To make a valid configuration it needed to provide a dynamic and runtime decision-making. To represent the domain knowledge and control strategy a uniform mechanism was used, and they embedded the knowledge about a single entity over several rules.

The knowledge maintenance was extremely difficult so a generic, domain independent model for the configuration task was suggested in the form of a constraint satisfaction, problem CSP, which will be described later [15][19].

3.2 Types of Configurators

A configurator is a software tool that can be used to help the customer or seller to carry out a purchase. The users are given a possibility to choose a product from different parts and their properties. The configurator makes sure that the chosen parts are compatible with each other [10].

The market of configurators can be confusing. There are many different software products that fall under the term configurator or product configurator. Unlike some other software, like accounting software, which has a more specific functionality, there is no such standard definition of what configuration software must contain [17].

The market is usually divided in to two distinct categories, sales configurators and product configurators. The term configurator is the broad term which describes both the sales configurator and product configurator. The term product configurator is often used when talking about configurators even when you mean both sales and product configurators, which can be confusing.
3.2.1 Product configurator

A product configurator is a type of configurator typically used internally at a company to verify a composition of articles to create a validated customized product. Basically, it makes sure that all the components in the product are compatible with each other. A product configurator is typically rather low-level and could be difficult to use by other than product experts. Its primary purpose is to help the user to combine and/or validate components in a complicated product. The output from a product configurator are typically a collection of the articles included (BOM, Bill of material) and a manufacturing order [13].

The term product configurator refers to a configurator with a focus on creating picking lists and ways of operation. The product configurator can from the finished configuration automatically generate a picking list of the material needed for the manufacturing and the ways of operation the machines need to produce the product [10].

3.2.2 Sales configurator

The term sales configurator refers to a configurator with a focus on creating an order. The sales configurator uses the made configuration to create complete order documents for the order department [10].

When used separately, a sales configurator produces a complete document for the order. Pricing and order status are typically functions of a sales configurator. A sales configurator also uses marketing rules to offer special option packages and package discounts. Sales configurators are proven to greatly improve the ability to sell and serve customers of complex, highly engineered products [18].

A sales configurator is a type of configurator allowing user-friendly questions and is typically used by internal and field sales staff, and on the web. A sales configurator has more functionality than a product configurator. It solves the same problems as a product configurator. The difference lies in the more advanced logics of the sales configurator allowing it to ask questions such as "What will the machine be used for?" and from these types of generic questions a configuration can be created. A product configurator would ask "Which type of machine do you prefer (Machine X, Machine Y etc)?".

Sales configurators are common on the web where user friendliness is of great importance. User friendly functionalities such as informative conflict resolution, ability to answer questions in arbitrary order and customized report generation are typical for a sales configurator [13].
3.3 Features of the Configurator

The **knowledge base** is the main technical component and consists of the **database** and **configuration logic**. The **database** contains the total set of components types and their instances. The **configuration logic** specifies the set of restrictions on how components can be combined [3].

The knowledge representation of a product configuration can be approached in several different ways. Below is a classification of conceptualization of configuration knowledge approaches [3]

- Rule-based approach
- Model-based approach
  - Logic-based
  - Resource-based
  - Constraint-based
- Case-based approach

3.3.1 Rule-based approach

Rule-based configurators work by executing rules with the following form: “if condition then consequence”. The product solution is derived in a forward chaining manner.

Forward chaining is one of the two main methods of reasoning when using inference rules (in artificial intelligence). Forward chaining starts with the available data and uses inference rules to extract more data (from an end user for example) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true. When found it can conclude, or infer, the consequence (Then clause), resulting in the addition of new information to its data.

In Rule-based configurators at each step, the system examines the entire set of rules and considers only the set of rules that can be executed next. Furthermore, there is no separation between direct relationships and actions. Thus rules contains both the domain knowledge such as compatibilities between components and the control strategy that is necessary to compute the solution to a specific configuration problem [3].
3.3.2 Model-based approach

The most important model-based representation that is implemented within configurators is logic-based, resource-based and constraint-based which will be described below [3].

3.3.2.1 Logic-based approach

The Logic-based approach is based on description logic. Description logics are formalisms for representing and reasoning with knowledge. They are based on the notations of individuals (objects), concepts (unary predicates, classes), roles (binary relations) and constructors that allow complex concepts and roles to be built from atomic ones [3].

3.3.2.2 Resource-based approach

The resource-based approach is based on a producer-consumer model of the configuration task. Each technical entity is characterized by the amount of resources it supplies, uses and consumes. A product configuration is acceptable when a resource balancing is realized [3].

3.3.2.3 Constraint-based approach

In constraint-based reasoning components are defined by a set of properties and a set of connection ports. Constraints among components restrict the ways components can be combined. A restriction can forbid a combination of parts (Part A cannot be combined with part B) or can require a specific combination (Part A requires part C) [3].

3.3.3 Case-based approach

Case-based reasoning relies on the assumption that similar problems have similar solutions. The knowledge necessary for reasoning consists of cases that record a set of product configurations sold to earlier customers. The current configuration problem is solved by finding and adapting a previous solution to a similar problem [3].
3.4 The Tacton view of Configuration Tasks and Engines

In this section a paper from Tacton System AB, written by Klas Orsvärn, and Tomas Axling [2] will be described. In this paper they give a description of which types of configurations there are according to them and they also describe different types of configuration engines and what scope of tasks the different engines can solve.

3.4.1 Types of configuration tasks

In a product configuration task, you put a requirement specification into the configuration system and get an output that is a list of components in a configurable product line which together satisfy the requirements and is in conformity with the technical restrictions of the product line. A description of how the component instances should be connected can also be done. The output is called a configuration.

A configuration engine is a computer program that matches input requirements with a suitable configuration (configuration task).

It uses a configuration model which is a representation of relevant knowledge of the configurable product line [2].

In the report [2] they have identified three relevant kinds of configuration tasks:

- Features-and-options configuration
- Compositional configuration
- Network configuration

3.4.1.1 Features-and option configuration

The configuration consists of a pre-determined number of components, which can be compulsory or optional, and each module must be realized by one of many of the optional components. The combinations of the components are restricted, and if you choose one of the components you may have to choose between some extra components [2].
3.4.1.2 Compositional configuration

The following are distinguishing features of compositional configuration [2]

- The configurations amount of components of a certain type can vary a lot.
- To see to that a configuration is valid you have to make certain that the components can be connected.
- In the configuration there are different ways to connect the components that are all reasonable.
- How components are connected is restricted and these restrictions may depend on earlier choices of connecting components.
- Requirements are to large extent expressed functionally rather than in terms of component attributes.

Features-and-option configuration is a special case of compositional configuration, and is often used for part of the configuration of compositional product lines. Thus a configurator capable of solving compositional configuration is also capable of solving features-and-options, but the opposite does not hold [2].

3.4.1.3 Network configuration

Network configuration is compositional in the sense that the number of modules is not pre-defined. The distinguishing feature of network configuration is that the requirements are expressed by selecting nodes (“components”) and connecting them together [2]. Connections are made at pre-defined points. Some connections of nodes are not valid. A node may be a component, a class of components, or a configurable part. If the node is a class of components, a component selection must be made. Different alternative components may have different connection restrictions. If the node is a configurable component, it may in turn be configured according to features-and-options configuration or compositional configuration. Its parameter value assignments may depend on the network configuration. Network configuration may thus be used to configure a network of nodes where each node itself needs to be configured [2].
3.4.2 Types of configurator engines

Three common types of configurator engines can be distinguished.

3.4.2.1 Product structure with restriction table

Requirements are expressed by selecting optional parts and alternative components in the part-of break-down structure. Restrictions are represented in a data-base, with tables representing valid, invalid required combinations of (usually) component pairs. The configuration engine assists the user by making sure that selections obey the restrictions, and by selecting required components [2].

3.4.2.2 Constraint solver

A constraint solver engine represents the configuration problem as a set of variables, which represent either requirement parameters or part of the configuration. Given requirements expressed as value assignments to some parameters, a constraint solver engine represents the solution as the domain of valid values of the other variables. One advantages of a constraint solver engine is that it offers a high degree of interactivity. Consequences of one parameter value assignment are directly propagated to other parameters. However the degree of interactivity differs between different constraint solver engines [2].

3.4.2.3 Constructive search

A constructive search engine will search for one suitable configuration that satisfies the requirements. The engine gradually constructs the configuration solution as it searches. With a constructive search engine, it is not feasible to select provably optimal solutions, since it would require all possible solutions to be generated and compared. However it usually allows local optimization, i.e. optimization for a sub-problem, either in the selection between candidate components, or for a larger part of the problem. This may be sufficient to generate a suitable configuration, i.e. a sufficiently optimal solution [2].
3.5 A report showing some vendors of Configurators

The numbers of vendors for configurators is increasing ongoing. This also because the Internet as a market place is increasing and the configurator solutions is helping to facilitate the order process on the web for both customers and sellers.

There are many different configurators on the market and finding out which configurator is the best for your company can be a difficult task

Reading a market research where many different vendors are tested against some criteria can be a start.

We have read two different market reports, one is a summary of a report from the consulting company Gartner from November 13th 2007 which shows different vendors configurators tested against some criteria and divided in classes of compliance [9].

The other research is from AMR Research December 2009 [8].

The report is mainly a study of 80 manufactures and interviews of 40 organizations about configurators. In our case the interesting thing in this study, is that they have tested 16 different vendors of configurators capacity against 4 criteria:

- Possible to deploy the configurator to SaaS(software as a service)/hosted deployment
- Have 2D and 3D visualization capabilities.
- Proven capabilities for engineered products
- Have online, web, and mobile deployment options for B2B and B2C

<table>
<thead>
<tr>
<th>Companies that fully satisfy all the criteria</th>
<th>Companies that fully satisfy 2-3 of the criteria</th>
<th>Companies that fully satisfy one or less criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Commerce (Camelion)</td>
<td>Cincom (Acquire)</td>
<td>Oracle (Oracle E-Business Suite)</td>
</tr>
<tr>
<td>BigMachines</td>
<td>Consoma (Configuration Solutions)</td>
<td>Oracle (JD Edwards EnterpriseOne)</td>
</tr>
<tr>
<td>Configura</td>
<td>Exact</td>
<td>Oracle (Siebel)</td>
</tr>
<tr>
<td>Configure one (Concept Enterprise Product Configurator)</td>
<td>FPX (Firepond onDemand Solutions)</td>
<td>SAP (Product Configurator)</td>
</tr>
<tr>
<td>Tacton Systems (Tstudio, TServer, TCsite, TCnomad)</td>
<td></td>
<td>Selecta (Configurator)</td>
</tr>
<tr>
<td>TDCI (BuyDesign)</td>
<td></td>
<td>Sterling Commerce (Sterling Configurator)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technicon Interactive (CustomCommerce)</td>
</tr>
</tbody>
</table>

Table 1. A summary of what level of criteria the different configurator vendors satisfy.
4 Comparative study of two Swedish Configurators, Tacton and COMBINUM

4.1 Background to the study

One of the tasks Logica requested was a deep study of the Tacton sales and product configurator. Tacton System AB is a company that develops a top configurator, seen in different market research and benchmarkings [8][9].

Because of various reasons we did not have the possibility to get a Tacton configurator license. This meant that we needed to get the information in another way. Klas Orsvärn at Tacton System AB were willing to have a meeting with us were we could interview him and he could tell us about the company and the Tacton configurator.

We were also in contact with another Swedish developer of configurators, In-Process, who sells and develops the COMBINUM configurator. They were kind to provide us with the possibility to test their configurator as an end-user and also to test their rule and constraint modeler, their module architect. Though this was a good help for us to understand configurators it was not enough to answer all our questions so we asked them for a meeting and interview. The interview questions for the two companies can be viewed in the appendix A and appendix B.

*The interviews were recorded and the information received from this has been put together in the report as they were answered. This means that all statements in the interviews and claims are from the persons interviewed.*

4.2 Tacton Configurator

4.2.1 The company Tacton systems AB

Tacton System AB is a Swedish company which develops and sells the Tacton configurator and performs service and consulting around this product. The Tacton product originates from research performed in the Swedish Institute of Computer Science, SICS, and Tacton System AB is a spin-off company from this. The research of the Tacton Configurator began in 1992.
4.2.2 Different parts of the Tacton configurator system

The Tacton system consists of **TCserver**, **TCstudio**, **TCsite** and **TCnomad**. The correlation between the applications TCserver, TCstudio, TCsite can be seen in figure 2.

**TCserver** is the application where the actual configuration is performed i.e. where the configurator engine is situated.

**TCstudio** is a graphical interface for facilitate the management for product model and product rules. It has a point and click system for making rules and constraints, but the system also allows the user to writes these from the keyboard.

**TCsite** is a user interface provided by Tacton System AB. It is a web application that can be customized and integrated into any other system as needed. It can be used by internal sales staff or an end-costumer. What the user get access to depend on its profile. For example profiles have different language, price lists, view or change saved carts, price, cost, discount, etc and the appearance of the application can change without programming using CSS (cascading style sheets).

![Figure 2. The Tacton configurator system.](image-url)
**TCnomad** is a windows application so the configurator can be used when it is disconnected from the configurator server. What must be done for this to work is that the computer in question has to have the configurator engine and TCnomad installed, and downloaded the latest model from the Tacton system, this in order to be up to date. When all of this is done, the user can create BOM (Bill of Materials), quotas and more. You can even send the configuration file to another employee to let him continue the quotation or put it back into the system se figure 3 for an illustration.

In appendix C there is an example application that uses the Tacton configurator.

![Diagram](image.png)

**Figure 3.** Illustration of the use of TCnomad.
4.2.3 The interview

As mentioned before the answers from the questions are presented in the way they were answered from the persons interviewed.

The questions we used in this interview are specified in APPENDIX A.

4.2.3.1 What kind of limitations does a configurator have?

The limitations vary greatly between different configurators.

The limitations a configurator should have, is that it is only supposed to solve configuration tasks. They are limited to solving a narrow class of configuration tasks.

The configurators that are more limited often have a way to go around its limitations, like the possibility to call another application that is written in another programming language. It is far from all configurators that have this ability, but it is quite common.

Poor configurators can work around their limitations through calling a programming language from its set of rules. If it is possible to have access to a programming language there are no limitations, only the matter of time and effort you want to spend in order to go around the limitations. One of the purposes of a configurator is to avoid programming.

The rules for the configuration should be expressed explicitly in the system. The purpose of configurators is that the rules should be able to be expressed explicit like data in databases.

Regarding the Tacton configurator there are no limitations when it comes to configuring products. If advanced calculations is needed that are not rule based you can write an application that you connect to the configurator.

4.2.3.2 What interfaces are there against other systems and how complex and flexible are they?

Tacton have clean and flexible interfaces.

In the Master data interface the configurator can retrieve data directly from other systems live when it is configuring. The advantage with this is that the data in other systems e.g. business systems will not be stored in two places.

The rule engine does not contain a database, i.e. the configurator does not demand a database. Tacton can provide ready-to-use applications that contain databases.
**Presentation layer interface** is how the user interfaces or the configurator interface looks like against the user interface.

The user interface can look in exactly any way you want and there are no limitations on how it can look like. It is a XML-based interface where the configurator describes what parameters the configurator task has. Metadata can be put on these parameters that show for example how they should be grouped. It is also used for the presentation layer which could be something completely different.

Tacton does not have any requirements on how the presentation layer should be coded.

Tacton provides an Out-of-the-box ready presentation layer. It is possible but not necessary to make your own presentation layers.

The **Configuration results** from performed configurations can be saved in XML format were you want. However, if the configurator is going to reconfigure the result, it has to recreate the XML format so the configurator can read it and continue in the same place as before.

The results of the configurations are flexible due to the fact that the configuration result is XML-based, it builds on the product model defined in the Tacton studio. This means that the configuration results can include any information at all, there are no limitations since metadata can be put on the product model in order to mark up the information, so that one can get what one wants in the system that are needed later in the workflow.

For example, if one shall integrate the configurator with a business system that needs to have work operations connected to its components in the configurator, which is quite common when the production planning is done, one needs to know which machines shall produce the products. This sort of information can be put in to mark up as one’s Business system wants it to be or if one wants to include CAD-parameters or something like that or can choose how it should be represented. In this sense the configurator could be said that it is flexible.

Tacton Systems sell ready applications. This means that you don’t have to do anything, but it is possible to change the appearance of the interface of the application. This in combination with flexibility and user friendliness which runs through everything.

Research and development on the configurator was performed for a long time before they started the company Tacton System AB which gives strength compared to how the other competitors have had to work.

The response time does not increase with increase of simultaneous users, because Tacton uses load balancing with help of a “configuration farm” which can contain any number of processors. In this way any number of simultaneous users can be handled without affecting the performance.
How bandwidth intensive the configurator is depends on the design of the end user interface. There is no inherent bandwidth of the configurator.

If you are an end-user and use the web browser, the only thing that is sent back and forth is html. With the presentation layer completely separate from the configurator, you can trim the communication the way you wish. The html pages can be constructed so that updates will be performed with very little information. With modern web technology, there are many solutions that make it so that very little information is sent back and forth to update it, once you built the page.

The Tacton configurator works well on bad connections.

Standard applications that are supplied are not optimized for bandwidth. The configurator sends the parameters you can see back and forth. If you are on a bad connection it can take a while to update but this does not depend on the configurator.

The Tacton configurator is CPU intensive.

The Tacton configurator can be run locally. The configurator engine can be installed locally on a computer as well as on a server. This is because it has a modular architecture that is platform independent and independent of the end-user interface. You can either have the same web application locally on the computer, if it is a web application that you run, or you can have a Windows interface on the desktop computer that works on the same configuration engine that would otherwise work on the web-based interface.

Tacton System can provide an application TCnomad, which is a Windows application that uses the same set of rules and the same rule engine as the configurator but with a different end-user interface.

4.2.3.3 How is the data handled? Interfaces with other systems?

The Master Data Interface handles the connection to databases.

Tacton is independent of how the other database is built, it does not even have to be a relational database. The Master Data Interface is actually an application you can write yourself. The configurator calls a program that retrieves data from somewhere and returns it to the engine.

The program can call any database, it could be a relational database using SQL calls, or it could just as easily make an API call for example if you have a PDM system or ERP system that do not recommend or allow access to its data with SQL calls.
4.2.3.4 Is it true that high flexibility means high complexity?

According to Klas Orsvärn it is true that high flexibility means high complexity when it comes to configurators. It has been very important from the outset for Tacton Systems to allow high flexibility without increasing complexity. Tacton Systems has accomplished this by a lot of research to understand what the right level of abstraction for product modeling is and the right interface to users and to other systems, for solving different types of configuration problems.

If you instead start with a limited configurator and gradually add supplements to make it more flexible, this increases the complexity a lot.

How complex it is to maintain a system in operation with changes and new products, depends on the product’s own complexity.

One of Tacton configurator’s advantages in the market compared to its competitors is that it is less complex to make updates in than other competitors systems.

For example: a customer reduced its rules with a factor 1000 with the Tacton configurator compared to its earlier configurator. Their maintenance is now simple; they constantly modify the data but the rules almost never change since the rules are defined in such a general level. Although the product is constantly changing the definition of rules is not changed that often.

4.2.3.5 How complex and flexible is the user interface?

Tacton Systems has focused on their user interface in order not to make it complex but still flexible. During research on configuration, before they developed the configurator, they identified what is important for a user when configuring. One important thing was that the user should be able to make his choices in the order he wanted.

Many configurators require that the user must make choices in a predetermined order. The fact that there is dependence between choices, the sequential dialogue leads to the users having to commit to a choice that may limit a selection in a later stage which is more important for this particular user. Then it would be better for the user to instead begin with the choice that is most important to the user and then make the first choice and see what options are valid.

With the Tacton configurator, you can make choices in any order but the whole time see which options are valid on the choices you have left and what is not done, which is a big advantage for the user. There’s no gain in having the possibility of making choices in any order if you cannot see how it affects the other choices.
When you configure it is always important that you can see how the number of options is limited by the choices that are made before. This can be done regardless of in what order the choices are made.

When it comes to a situation when you are interested in an invalid option, it may be that you are prepared to compromise choices made earlier. Then you can select an invalid option to see which options are valid or not. You choose an invalid option and then you receive a proposal on what to change in the previous selections in order to get this option. According to Tacton Systems this is a strength that many other configurators lack.

One other thing is that you don’t have to make all selections in the configuration, the configurator optimize over a prize, performance, etc depending how the configurator is instructed. This instruction can be decided by the configurator or by a choice from the user. Then the configurator can propose solutions on the parameter which has not been specified.

In the TCstudio you can define how you want to set up the user interface. In most applications it is enough to define how the parameters are grouped and what kind of parameters there should be.

There is only one way of setting up the interface but you can put metadata on the parameters in order to tailor your interface for example to decide which position a field should have.

4.2.3.6 How user-friendly is the configurator?

Tacton Systems has focused on making it as user friendly as possible

4.2.3.7 Are there different types of complexity as the logic, user interface and/or API?

No information was received under this question mostly due to that Klas Orsvärn did not understand the question.

4.2.3.8 How does the modeling of the object that will be configured look like?

The modeling is done in an application called TCstudio (Tacton studio) that is divided into different views. The views are divided into:

- Component view
- Configuration view
- User view
The configuration view is the most interesting view, because that is where the modular product structure is defined, i.e. how the product is divided into modules. All configurable products are in some sense divided into modules. It can be a good or bad modularization but there is always a module subdivision. One can define it as you want. The modularization has a hierarchical structure.

In the structure you also add the rules and properties. The hierarchical structure that you define in a conventional manner, as an Explorer-like view where you can add constraints (rules). These rules of the modulation language are designed so that the rules are very powerful, but you cannot get away from that somewhere there must be rules. These rules are expressed by means of logical conditions that you define. There is a rule editor where you can point and click to build up the conditions based on its modular structure.

If one wants to be fast, you can work with the keyboard, and you then get completions i.e. it recognizes what you want to write and you just need to press Tab to make it clear. For the most part, however, it is a clean graphical interface structure.

Most maintenance is clean product data maintenance that is pure table type of information. Tacton Systems is doing its utmost to reduce the amount of rules, in order to make the system better.

4.2.3.9 At what level of abstraction is the modeling?

In the Tacton configurator the abstraction level of the modeling lies as high as possible.

The modeling is done in the module TCstudio which is a link between the company’s product information and the Tacton Configurator Server.

The most part of the model is data and can be made in Excel or any database. The rules can be made in XML-format. This means that the rules can be made in Excel or from other configurators as long as it has the correct XML-format.

4.2.3.10 What kind of support can you get from the software provider?

Tacton Systems AB provides conventional software support on the Tacton Configuration systems. They can also provide standard applications. Tacton Systems performs consulting service around the Tacton configurator system.
4.2.3.11 What are the licenses costs?

The licenses cost depends on how many concurrent users you want to have for your configurator. For obvious reasons Tacton Systems doesn’t want to show their prices because of competitors. Also the prices is a discussion matter with new potential customers.

The price level for this kind of complex software is not low and can be up in the millions depending on the how complex systems you have.

4.2.3.12 Some other questions

In the integration where do you spend most time and in which module?

This depends on the other system you are making integration against.

How does the maintenance work after implementation?

You need to have maintenance of the data, rules and the application.

Is it possible for the customer to make the maintenance of the system himself or does it need some continuous expertise help from Tacton?

The customer can handle the maintenance themselves but it is more efficient if Tacton System does it for them.

If the customer can handle the maintenance themselves what skills does this require?

The data maintenance does not require high skills but he required skills for the maintenance of the rules depend on how complex configuration tasks you have.

Is Tacton always the one who implements the system or have you had clients who have made it themselves?

There are clients that have made the installation themselves.

How can you connect to the Cloud and Silverlight?

It is possible to install the configurator on a cloud type like Amazons server farm.

Tacton Systems have had cooperation with salesforce.com (cloudbased) for the end user application that can run the configurator and end user interface but not the rule engine. This is because salesforce.com can only run AJAX-applications.

The connection to Silverlight should work because the presentation layer is separated from the configurator engine. So it should work but there is no verification on this.
4.3  COMBINUM Configurator

4.3.1 The company In-process

In-Process Sweden AB is a small Swedish software company situated in Borås. In-process Sweden AB sells and develops the Combinum configurator and has long experience of configurators. They have developed every kind of configurator (Sales support, Article structure, graphics and 3D).

4.3.2 Different parts of the COMBINUM configurator system

COMBINUM consists of different modules and COMBINUM Sales Configurator is the base module that contains COMBINUM web server and COMBINUM Architect.

COMBINUM Architect
Is an administration tool for maintaining the rules and product model. It also has a test environment where the administrator can make changes and test them without changing the live system.

COMBINUM WebServer
Is a web based application that is used for configuring products.

You can add on other modules on the base module in order to get more advanced functionality that is not supported by the base module.

The other modules are:

COMBINUM Advanced Pricing
Gives advanced options for price handling as:

- The price is calculated automatically as the user sets the product's parameters.
- Price calculation with complex price rules and formulas
- Hierarchical price dependencies for efficient handling of dimensionally ruled prices
- Multiple price lists
- Perspicuous price reports with sub-totals and notations
- Discount rates
- Exchange handling
COMBINUM Advanced Product Data

Gives advanced option for product data as:

- Advanced rules for when a product data should get a certain value
- Formulas for calculating the value of a product data
- Easy to use matrix for fast entry of multiple product data controlled by multiple parameters

COMBINUM Composite

Makes it possible to configure systems where each component is configured separately.

Some applications for this are

- Automation line with machinery and intermediate carriers.
- Sections of hydraulic valves in which each section is governing a functioning.
- Computer rooms with server racks with servers and other components.

COMBINUM Graphics

It is an advanced graphics engine for displaying an image of the configuration. The image evolves when the user makes changes to the configuration. The graphics engine is generating the graphics in a web browser and can use Microsoft Silverlight or scalable vector graphics for it.

COMBINUM Multi-Language

It is a module for making the configurator available in multiple languages. All the variables and texts in the web configurator can be translated and all the data the administrator is putting in. COMBINUM supports an unlimited number of languages.

Some other featured items for facilitate the Management of multi language:

- Define which fields that should or should not be translatable.
- Export and import translations to Microsoft Excel for easier maintenance or for sending the spreadsheet to a translator

COMBINUM Nomenclature

Is a module that generates a product key automatically and it gets changed as the configuration proceeds. A product key is a string of text that describes the configuration. This product key is automatically updated during the configuration.
COMBINUM Reports
This module lets the user create Microsoft Word reports like quotation letters and order acknowledgements. And then automatically fills it with the right information from the configuration design.

COMBINUM SmarTeam Integration
Is a module for integration with the PDM system ENOVIA SmarTeam. The integration provides possibility to publish documents from SmarTeam directly into COMBINUM.

COMBINUM Structure
COMBINUM Structure is the module for management of item and bill of materials. Bill of materials (BOM) and routings are generated automatically by COMBINUM Structure.

In APPENDIX D there is an example application that uses the COMBINUM Configurator.

4.3.3 The interview
As mentioned before the answers from the questions are presented in the way they were answered from the persons interviewed.
The questions we used in this interview are specified in APPENDIX B.

4.3.3.1 What kind of limitations does a configurator have?
With the type of rule set that exist in the COMBINUM configurator to deal with what can and cannot be combined, this rule set is of relational type. It means that this component may not be combined with that component. It is built to create a user-friendly interface so you do not need to use formulas when you have simpler conditions.

Such rules, when you have a formula, such as total mass of all its components shall not exceed 1000 kg. Such regulations cannot cope with pure combinable. For example you may have one thing combined with another but if we have them all it may be too heavy, then you need a written rule to solve this.

COMBINUM has now come so far that it has these items also by what they call Attributes. An Attribute is put up so that you, with formulas, can make calculations. For example, if it turns out that this goes over a limit you can send a message to users telling you that something has been done that you should not do. COMBINUM have had these restrictions before.

The rule set can be divided into two groups constraint-based and relational-based.
COMBINUM have in one part of the rule set relational-based rules that provide greater ease of use but they are also able to use constraint-based rules when needed.

COMBINUM does not support the possibility to generate 3D graphics in the configurations. But it is possible to develop this in the future, if they get a client that wants them to provide it. The knowledge exists in the company, because they have earlier develop the same to ABB Olofström connected to Solidworks 3D graphics, but this was not for the COMBINUM configurator. COMBINUM cannot make planogram configurations (setting objects in the room) because the COMBINUM configurator does not support configuration with 3D graphics.

4.3.3.2 What interfaces are there against other systems and how complex and flexible are they?

Export of data
COMBINUM support XML exports of generated article structures. XML exports generated configurations, values that have been chosen etc. To build an XML export does not take long, so they can quickly build the report the client wants.

Import of data
For example, prices that are coming from another system depend on what you have done in COMBINUM to handle the price. The price may be linked to the features or articles, depending on how the client has divided it.

On the article side, you can make a plug-in that read information and take in the data that concerns the articles, in real time.

Although COMBINUM supports article structures, generating BOMs rules-based you do not need to store the articles in COMBINUM if you want, this due to that one often want to avoid duplication of storage. Then you can read the article structures on-the-fly from a business system and combine it with the rules of COMBINUM. But of course a small plug-in must be written for each business system how to retrieve the articles, because it is different for every business system. The client could make the plug-in but knowledge of programming is needed.

For example, if you want to create an order or order lines when connecting to an order system or CRM system, and then select the product and expect to enter the configuration mode to configure it. The configurator is prepared so that from this mode can come directly to the configuration page and skip the logon, and more things that are associated with the CMR system – in order to keep it as seamless as possible.

Programming is .NET right through the system.
COMBINUM distinguishes all adaptations from the base system.

You can change all business logic and all layouts, etc. without changing the base system and hopefully the adaptations will keep up with the updates, but In-process cannot promise anything, it is not always the case that there is no adjustments necessary on the adaptations.

Possibility of customization is the same as building the system, adaptations that has been made have later been incorporated in the base system.

For example the module Structure is built in the same as a customization, but is shipped as part of the system.

To make customization on the system you have to have knowledge on programming in C# and some help in the beginning from In-process.

When users make heavy configurations, for example when you make a selection and check in anything then some other choices gets invalid due to this i.e. it gets grey, then the graying of these invalid choices should not be 10 seconds afterwards, but if the price is delayed a little bit before it is showed it is not so important. So the priority for users is to get feedback to the user interface. COMBINUM has solved this by using threads, two different priorities on these threads for fast response back to the user even if it is a queue to the server.

When it comes to if COMBINUM’S scalability and whether it is a linear or exponential it is neither.

COMBINUM does not read the information directly from the database on how the product is stored. It is smart and polls its resources. The information is instead read and put in the cache. COMBINUM can then have a fast interchange with the cache instead and it is created as many caches as it believes it needs to reuse the information stored in the caches later.

The first users take more resources than the 10th user for then start to poll the caches and reuse them. This make the scalability more being a digressive curve rather than a linear or exponential.

COMBINUM uses a serialization that is known for its compactness so it sends as little information as possible.

The system has been tested to run on bad connection and it works (It was not said how well).

The system can be calculus intensive if you want, it depends on how complex calculus you have, but it doesn’t have to be.

The whole COMBINUM can run as a stand-alone client, even on a lap-top. The challenge with offline-solutions is the synchronization back to the mother database. In-process have a solution on this but will not make this before they have a demand from a client.
4.3.3.3 How is the data handled? Interfaces with other systems?

A database is use that can either be Microsoft SQL Server or ORACLE, but SQL server it preferred.

A layer between the database and the object is used when moving data, an object server. This avoids writing SQL-code to fetch and handle table rows, this means that you work directly against the objects.

To work against the other information sources in a company or other databases plug-ins are used which must be written specially for the connection you want as discussed before.

4.3.3.4 Is it true that high flexibility means high complexity?

COMBINUM is built so that programming should not be needed for the user. With flexibility you can create so it gets more complex if you want to, but the main thing is not to make it more complex than necessary.

It is possible to have an administrative site where the database is divided into a production environment and a development and where you have a test environment in which to write new rules that can be tested before implementing it in the production.

There are replication capabilities from test environment to production environment, you can decide when you want to take the new changes into service and then they will be replicated from test to production environment.

4.3.3.5 How complex and flexible is the user interface?

If you want you can make it complex but the aim is to make it as user friendly and flexible as possible. The proof of user friendliness is that the customers that pick them as a supplier does so because of this aspect.

4.3.3.6 How user-friendly is the configurator?

In-process claims that COMBINUM is user friendly. The feedback they have had from the customer indicates that the user friendliness is a big reason why they have been given the contract and also that they are experienced to be flexible as a company against clients.
Are there different types of complexity as the logic, user interface and/or API?

The user interface for end users is web interface which means that there is no support for mobile applications, but only in web browsers.

How does the modeling of the object that will be configured look like?

COMBINUM Architect which is included in the base system COMBINUM Sales Configurator is a tool for managing and maintaining the configurators rule set.

At first we have a product tree that looks the same as on the web. The products are divided into families. You can have a product that consists of a product. This in turn has to do with the fact that a configuration sometimes contains another configuration.

For example, when you configure a door and the door has a window in it that is a configuration itself. This means that you have configurations in two levels.

In this way you work with the tree throughout the process. Once you have reached down to the product, you have to choose which tree you want to expand in order to watch.

If you want to look at the interface elements to add new tabs, parameters, images, text, etc., then it is a subset. Another subset can be expanded to deal with interface rules, handle the article structure, price, nomenclature, text etc. So it is divided so you don’t have to handle all elements at once. The challenge is to not make it too complex for users.

There is no special language involved only logic (and, or - standard operators) you can add functions. These functions can be added as customer adaptations without rewriting the rule engine. For example, if you need the sine function and it is not available you can add it as an adaptation, when it says sine the argument should be received and returns this result. You could say that the rule engine is extendable at several levels, partly by In-process and partly by customers and retailers.

The feedback received from customers is that the user friendliness is the cause for them to get the contracts. A customer should be able to administer the models and rules.

At what level of abstraction is the modeling?

In the COMBINUM Architect the modeling is done with the abstraction level of the modeling as high as possible.
4.3.3.10 What kind of support can you get from the software provider?

In-Process can offer support contract which includes getting new versions of the system, bug fixes and help if the system does not work.

Normally the adaptations are distinguished from the support contract, all that is related to customer adaptations are charged ongoing. If an upgrade makes it necessary to rewrite a customer adaptation, is this not included in the support contract. However, they make it known to the customer that an upgrade may involve changes in customer adaptations.

The customers can, against a higher fee, get the adaptations included in the contract, but it is difficult for In-process to include this in the price list when they do not in advance know how much adaptation a customer need.

4.3.3.11 What are the license costs?

Consulting fee that consist of project handling, education of the customer, start of the system in order to see that everything works. The consulting fee and License fee varies between different project. Big organizations that have bigger licenses tend to have more need for support from consultants.

COMBINUM have a module structure. You have the base system which you then add the modules you need. This means that you pay for what you need.

The license fee depends on how many “Named users” that are active in the system or if you have anonymous users you can pay for how many configurations that are made in a year. The license fee varies and is always set for the whole system depending on what modules you have, where the number of administrators, users and configurations are measured.

4.3.3.12 Some other questions

In the integration where do you spend most time and in which module?

This depends on what you want to accomplish with the integration.
It could be in the module COMBINUM Structures were the article and article structures is handled. In the base module sales Configurator were you adapt to the web interface.
In the administration tool COMBINUM Architect you can make functions.
**Where is the most effort put in when you configure?**

It depends on the customers products.

It’s not hard to implement the system, this mostly depends on what the customer wants and how their situation is.

Installation with documentation takes 1-2 day but then the customer need to have prepared an installation of the SQL-server.

**Where in the integration is most complex?**

It’s in the module Advanced Product Data.

**How does the maintenance work after implementation?**

It’s so user friendly that the customers can handle it themselves.

**Is it possible for the customer to make the maintenance of the system himself or does it need some continuous expertise help from In-process?**

The customer can manage the system themselves and need no assistance once the system is up and going.

**If the customer can handle the maintenance themselves what skills does this require?**

The data maintenance does not require high skills.

The required skills for the maintenance of the rules depend on how complex configuration tasks you have.

**Is In-process always the one who implements the system or have you had clients who have made it themselves?**

They have always done the installations for their clients.

**How can you connect to The Cloud and Silverlight?**

When it comes to connect to The Cloud, COMBINUM’S administration interface is not yet web based which has to be addressed before this can happen.

Silverlight works because it is now used in COMBINUM for the 2D graphics.
4.4  A summary of the information from interviews

Here will be described if we think the questions have been answered and the value of the answers and its limitations and other comments. A table that summaries the information received from the interviews will be shown in the end of this section.

4.4.1 Q1. Which configurators are there on the market?

There are hundreds or maybe even thousands of different developers and sellers of configurators. The figures are rising for every year. But to know which configurator that is the one you should choose, reading one or a couple of market research reports is one important way to go. We have seen two market reports one that is retelling of the report from Gartner [8] and a report from AMR research [9]. The vendors that are in the report are the leading vendors in this field and they are also ranked according some criteria that is described in the marketing report.

To answer the question, we have just accounted for the market report from AMR research in our report and not described the different vendors. This we believe is out of the scope of this report. One way to have made the report better is to have included a market report from the Aberdeen Group. This report shows the best in class configurators, i.e. it is a benchmarking report. But of course as all good information, it is a cost issue. The report was expensive. This is why we decided not to include it.

4.4.2 Q2. What kind of limitations does a configurator have?

In both interviews with Tacton Systems and In-process, they stressed that the configurator first of all is supposed to solve configurations problems and that the limitations varies between different configurators. The question is answered in the configurations section 3 and in the interviews. The limitations depend on the features of the configurator and a discussion in the interviews is also made how to go around the limitations.

Even if you have a poor configurator you can go around the limitations by calling a programming language from your rules. In this way you can make an adaption that solves your problems. It all depends on the time and money you want to spend to solve the problem.
4.4.3 Q3. What interfaces are there against other systems and how complex and flexible are they?

The report shows in the account for the interviews how the both configurators do to export and import data from the system.

From the interviews could be deduced that how complex and flexible it is to connect against the other system depends largely on other system you are connecting to.

4.4.4 Q4. Is it true that high flexibility means high complexity?

In all information we have read and according to statements from the interviews all stated that more flexibility means more complexity. They also said that their products are not complex but still flexible. What they mean must surely be that to make the software product flexible is complex. But for the user of the software the interface is not complex but flexible.

4.4.5 Q5. How complex and flexible is the user interface?

Both Tacton and In-process has focused on not to make the interface to complex but still flexible. This is for us little bit hard to verify completely. But from what we have seen in some test examples both Tacton and Combinum are easy to handle.

4.4.6 Q6. How user-friendly is the configurator?

This question is also depending on what perspective the user or the developer. Both the Tacton and In-process claims that their configurator is user friendly.

4.4.7 Q7. Are there different types of complexity as the logic, user interface and/or API?

This question was hard to understand for the persons interviewed and the answers were not conclusive.
4.4.8 Q8. How does the modeling of the object that will be configured look like?

Tacton configurator has the TCstudio application and COMBINUM has the Architect to model the products. They are used to make the modeling easier for the user. This to make it so you don’t have to have good knowledge of programming, it is more about that you have the knowledge of the product you are configuring.

4.4.9 Q9. On which abstraction level is the modeling?

Regarding this point, both Tacton Systems and In-process is striving to have the modeling at the highest abstraction level possible, this to make it as user friendly as possible.

4.4.10 Q10. What kind of support can you get from the software vendor?

Both Tacton Systems and In-Process have ordinary software support and the help they support is mostly depending on arrangements with the customer.

4.4.11 Q11. What are the licenses fees?

The license fee was hard to get figures in money terms. This is because both Tacton Systems and In-process don’t want to reveal any prices in a written report that can be seen by competitors and potential clients. This is of course understandable because the product they are selling depends more on the customer’s needs and what adaptations for his solutions has to be made. This is more a question of negotiations between seller and customers.
4.4.12 Summary of the information from the interviews in a table

Below can be seen summary of the information gathered from the interview. It shows a comparison between the two configurators.

<table>
<thead>
<tr>
<th></th>
<th><strong>Tacton configurator</strong></th>
<th><strong>COMBINUM configurator</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base module</strong></td>
<td>TCserver</td>
<td>COMBINUM Sales Configurator</td>
</tr>
<tr>
<td><strong>Platform</strong></td>
<td>Platform independent</td>
<td>Windows</td>
</tr>
<tr>
<td><strong>Modeling tool</strong></td>
<td>TCstudio</td>
<td>COMBINUM Architect</td>
</tr>
<tr>
<td><strong>CPU-intensive</strong></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solution, load balancing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distribute to several servers</td>
<td></td>
</tr>
<tr>
<td><strong>Connection against database</strong></td>
<td>All databases that support JDBC or ODBC.</td>
<td>MS SQL SERVER or ORACLE</td>
</tr>
<tr>
<td><strong>Interface against other systems</strong></td>
<td>Master data interface - applikation</td>
<td>.NET C# plugin for every system that you connect to.</td>
</tr>
<tr>
<td></td>
<td>Presentation layer interface - XML</td>
<td>Generated article structures – XML</td>
</tr>
<tr>
<td></td>
<td>Configuration results - XML</td>
<td>Generated configuration results - XML</td>
</tr>
<tr>
<td><strong>Possibility to run locally</strong></td>
<td>Yes, TCnomad</td>
<td>Yes, but the synchronization against the mother database is not entirely solved</td>
</tr>
</tbody>
</table>

*Table 2. Summary of the comparison of the two configurators Tacton and COMBINUM.*
5 Conclusions

5.1 Introduction

The main goal with our work has been to find information so that the questions Logica have about configurators could be answered. This has been done in another way than originally planned. The deep study of the Tacton configurator was cancelled due to lack of access to software and in any case it had not been a comparative study unless we also had been able to get another configurator to compare it with.

Instead the work switched from being a practical study of software with some potential programming, to more of a paper study with the interviews as our greatest source of information. In the end this probably gave us more information than if we had tested the products ourselves. The persons interviewed has long experience in the field of configurators and they shared this knowledge with us.

We were at least able to test the COMBINUM user interface and its rule and constraint administration tool, Architect. This gave us some insight on how to work with the rules and constraints in the configurator.

In all the information we have gathered it is hard to find negative aspects of the configurator. This probably has to do with that fact that it’s mostly developers and sellers that are the source of this information. It is not a good selling point to reveal the downside of your product. We may have lost some of our objectiveness when not testing the products ourselves.

5.2 Did we reach our goals with the study?

The first goal was to find out the advantages and disadvantages of different types of Configurator systems.

The advantages we found were:

- It cuts down the number of errors in the configuration process.
- Reduces the lead-times in the sales-delivery process.
- Efficiently distribute up-to-date product knowledge to the people in the configuration process.
- Increase the volume of quotations without increasing the number of sales persons and engineers.
The disadvantages:

- The configurator is complex.
- It is hard to maintain the rule set.
- There is no real standard for what to include in the configurator software.

The second goal was to find out how complex and flexible configurators are.

From the interviews and information from other sources such as papers and Internet, everything pointed to the fact that configurators are complex. How complex they are, depend mostly on how complex the product you want to configure is.

The third goal was to make a deep study of the Tacton configurator.

This we were not able to do, but instead we were able to interview persons that have long experience in the field of configurators. This was persons from two different Swedish companies that develop configurators. Tacton Systems, that develops the Tacton configurator and In-process, that develops the COMBINUM configurator.

The information we received from them gave us answers to our questions and an understanding of the field of configurators which we would not been able to understand without their support.

5.3 What can you do with the result of the report?

The result of the questions answered in the report does not lead to any recommendations. To give a recommendation we would have needed a background problem that someone wanted to be solved. In this case there are no real problem to be solved just the interest from Logica. The goal has only been to answer the questions and this has been done as described before. What the report gives us is an insight to the field of configurators and a stepping stone for a deeper study of configurator software.

5.4 How can the study go forward from this point?

A lot of information has now been gathered about configurators and the configuration process, in this report. And the next step is to make a comparative study with testing of the software.

Such a study would use a sufficiently big case of a configurable problem to be solved in two different configurators, in order to see the whole process from start to the finish, how rules and constraints are implemented, what needs to be done to go around problems etc. Also how the configurator system needs to be set up to work would be of interest.
References


[10] COMBINUM’S webpage (In-Process is the company name behind COMBINUM) who provides the COMBINUM sales configurator, was visited 23 April 2010. URL: http://www.combinum.se/


[13] Allconfiguratos.com webpage that provides information about configurators, was visited the 17 May 2010. URL: http://www.allconfigurators.com/


[16] BIGMACHINES’S webpage who is a provider of sales configurators, was visited the 7 April 2010. URL: http://www.bigmachines.com/

[17] Configure One, 2008, Whitepaper, Four Types of Configurators, Which One is Right for Your Business?, A buyer’s guide to understanding the configurator market., Oak Brook, USA.

[18] Cincom Acquires webpage who is a provider of sales configurators, was visited the 17 May 2010. URL: http://www.cincomacquire.com/

APPENDIX A
Interview questions Tacton configurator.

Questions used in the interview and discussion regarding the Tacton configurator with Ph.d Klas Orsvärn at Tacton Systems AB, Stockholm Sweden May 4\textsuperscript{th} 2010.

1 Problems and questions at issue.

1.1 What kind of limitations does a configurator have?
1.1.1 Are there limitations on which configuration solutions that can be performed?
1.1.2 Are there other types of limitations?

1.2 What interfaces are there against other systems and how complex and flexible are they to handle?
1.2.1 Does the complexity depend on which system one connects to?
1.2.2 How does Tacton handle scalability when for example going from 10 – 10,000 simultaneously users? Will it be a linear or exponential curve?
1.2.3 How bandwidths intensive are the configurator?
1.2.4 How does the configurator work when bad connections are used?
1.2.5 Are Tacton calculation intensive?
1.2.6 Is it possible to run locally and if so, how does it work?
1.2.7 How does Tacton’s interface work with other programs?

1.3 How is the data handled? Interfaces with other systems?
1.3.1 How does the connection to other databases work?
1.3.2 How does Access bean work?
1.3.3 How does integration package work?

1.4 Is it true that high flexibility means high complexity
1.4.1 How complex is it to maintain a system in operation with changes and new products?

1.5 How complex and flexible is the user interface?
1.5.1 How do you do to put up the user interface?
1.5.2 Are there alternative ways of setting up the interface?

1.6 How user-friendly is the configurator
1.6.1 The issue is from the developer point of view?

1.7 Are there different types of complexity as the logic, user interface and / or API?

1.8 How does the modeling of the object that will be configured look like?
1.8.1 How is the modeling perform?
1.8.2 What kind of configuring language is used when the modeling is performed?
1.8.3 How user friendly is the modeling?

1.9 At what level of abstraction is the modeling?
1.9.1 Do you need the TCstudio to do the modeling or can you do it directly in TCservern?
1.9.2 Is the TCstudio only used to facilitate?

1.10 What kind of support can you get from the software provider?
1.10.1 What kind of solutions can Tacton provide?

1.11 What are the licenses costs?
1.11.1 How much of the cost of software in system solutions?

1.12 Other questions
1.12.1 In the integration where do you spend most time and in which module?
1.12.2 Where is the most effort put in when you configure?
1.12.3 Where in the integration is most complex?
1.12.4 How does the maintenance work after implementation?
1.12.5 Is it possible for the customer to make the maintenance of the system himself or does it need some continuously expertise help from Tacton?
1.12.6 If the customer can handle the maintenance them self what skills does this require?
1.12.7 Is Tacton always the one who implements the system or have you had clients who have made it themselves?
1.12.8 How can you connect to the cloud and Silverlight?
APPENDIX B

Interview questions COMBINUM configurator.

Questions used in the interview and discussion regarding the Combinum configurator with Martin Nilsson CEO and Kjell Green Sales Director of the company In-process. The interview was performed at their office in Borås May 7th 2010.

1 Problems and questions at issue.

1.1 What kind of limitations does a configurator have?
1.1.1 Are there limitations on which configuration solutions that can be performed?
1.1.2 Are there other types of limitations?

1.2 What interfaces are there against other systems and how complex and flexible are they to handle?
1.2.1 Does the complexity depend on which system one connects to?
1.2.2 How does COMBINUM handle scalability when for example going from 10 – 10.000 simultaneously users? Will it be a linear or exponential curve?
1.2.3 How bandwidths intensive are the configurator?
1.2.4 How does the configurator work when bad connections are used?
1.2.5 Are COMBINUM calculation intensive?
1.2.6 Is it possible to run locally and if so, how does it work?
1.2.7 How does COMBINUMS interface work with other programs?

1.3 How is the data handled? Interfaces with other systems?
1.3.1 Does COMBINUM have its own database?
1.3.2 How does updates work?
1.3.4 How does changes against other companies information sources work?
1.3.3 How does connection to other databases work?

1.4 Is it true that high flexibility means high complexity
1.4.1 How complex is it to maintain a system in operation with changes and new products?

1.5 How complex and flexible is the user interface?
1.5.1 How do you do to put up the user interface?
1.5.2 Are there alternative ways of setting up the interface?

1.6 How user-friendly is the configurator
1.6.1 The issue is from the developer point of view?

1.7 Are there different types of complexity as the logic, user interface and / or API?

1.8 How does the modeling of the object that will be configured look like?
1.8.1 How is the modeling perform?
1.8.2 What kind of configuring language is used when the modeling is performed?
1.8.3 How user friendly is the modeling?

1.9 At what level of abstraction is the modeling?

1.10 What kind of support can you get from the software provider?
1.10.1 What kind of solutions can COMBINUM provide?

1.11 What are the licenses costs?
1.11.1 How much of the cost of software in system solutions?

1.12 Other questions
1.12.1 In the integration where do you spend most time and in which module?
1.12.2 Where is the most effort put in when you configure?
1.12.3 Where in the integration is most complex?
1.12.4 How does the maintenance work after implementation?
1.12.5 Is it possible for the customer to make the maintenance of the system himself or does it need some continuously expertise help from In-process?
1.12.6 If the customer can handle the maintenance them self what skills does this require?
1.12.7 Is In-process always the one who implements the system or have you had clients who have made it themselves?
1.12.8 How can you connect to the Cloud and Silverlight?
APPENDIX C
Tacton configurator, an application example.

This is an example of an application that uses the Tacton configurator.

This example shows how a customer can configure a truck after his the needs. The different components he has to make a selection on:

- Cab-type
- What engine and how powerful it need to be
- Class
- Chassi type
- Wheel conf.
- Chassi height
- Suspension

Figure 4.
This is an application that is built for the Bisontrucks company and to start with the selection of parts of the truck is optimized on the price which can be seen in the checkbox for optimize on is checked.
Step one in this example is to start with choosing the Cab-type. Normally you can start at whatever component to make your selection. In this example however we are performing the selections in sequential order. As you can see in figure 5 there are a couple of different cabs to choose from and the CR19T is chosen. The picture in Cab-type square is now changed to the picked CR19T cab and the checkbox lock is checked that indicates that you have made your choice.

Next step is to choose your engine and how powerful engine you need (engines hp). Some of the engines are colored white to show that they are not allowed to be picked in this configuration. If you try to pick this engine you get a warning of inconsistency. But it also gives a message which tells what you have to change in order to make this a valid choice of engine. In this example we have to the cab from CR19T to CP14.
If horse power 530 is chosen all other engines are whited out except the D14 engine.
The rule and constraint that work under the engine selection part give us that there is a
correlations between engine and horse power.

Then what type of class the truck will have has to be picked which depends on what the truck is
supposed to carry. The Long haul is picked out of the alternatives in Figure 8.
You then choose between single or double bogie, double is chosen.

In figure 10 all the possible wheel configuration is shown and we pick 6x2*4. In this configuration the last wheel pair has steering capacity.
Different chassis heights are possible but we chose Normal. As can be seen Extra low and high is not possible choices.

Last we have the suspension on the truck and chose full air of the two choices on hand in the figure 12. Now all the parts have been chosen and we have a specification on the truck and a price 902620.
When we started the configuration before the configurator had made a choice for us based on price. If all the lock check box is removed we can see in the figure 13 that we have let the configurator make a configuration optimized on comfort. The configurator has used rules and constraint to pick part that will make the solution for us. Optimization based on Versatility, Performance and Fuel economy can also be done.
APPENDIX D
COMBINUM configurator, an application example.

The company Pipelife sells material for various solutions of wells. This example will show how to configure a descendent well with help of an application made with the COMBINUM configurator.

In Figure 14 above, in the blue rectangle, there are different well types that you can make selections from. The green circle under the picture of the different well types shows that they are valid choices.

In Figure 15a, a selection has been made, the well that has been chosen is the descendent well. As soon as we have made our selection the other choices will be invalid. This will be indicated by the circles turning from green to reed with an x inside it.

The configurator has also chosen the well dimension 1000 mm for us which is built in the application with rules.
In the bottom of the screen, figure 15b - in the same step as in figure 15a, - a list of the articles on the material that is needed for assembling the chosen well. The list of articles is based on the selections you have made so far. The list is appended with more articles as you go along and make your selections. The configurator figures out what articles are needed and how many of each. This is based on the rules and restrictions that are decided before hand in the application.

The next step in figure 16, is to chose the dimension of the connecting pipes to the well. Depending on what dimension is selected you get different sets of choices to make selections from. This can be seen on the right side of the pipe dimension selection.

When the configuration is finished you get the total price and a specification on what articles you need for the well and how many, see figure 17. A separate specification can be created which for example can be put in a document for an offer, see figure 18.