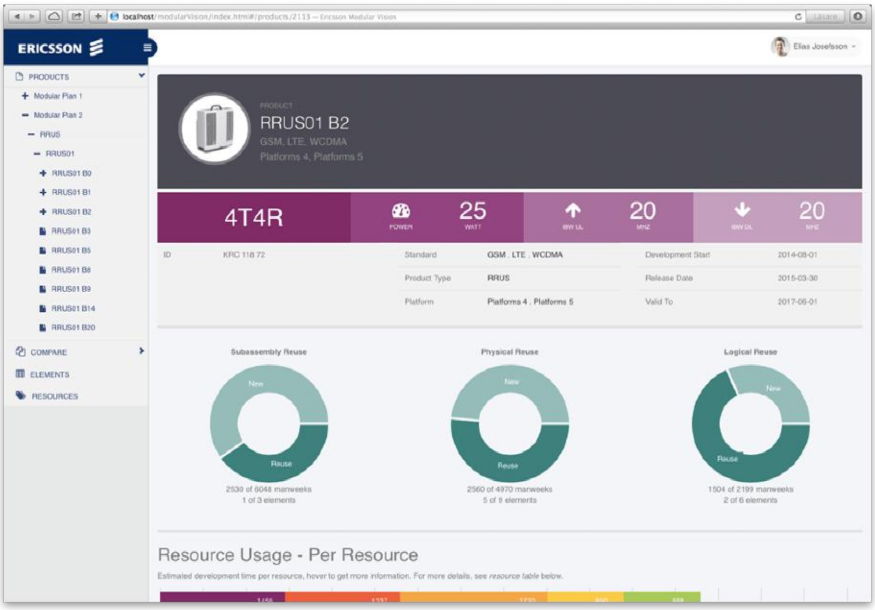
Multi-viewed Visualization of Modularity for Product Line Management

Case study at Hardware Radio Design, Ericsson AB

Elias Josefsson



Master of Science Thesis MMK 2014:60 MDA 471

KTH Industrial Engineering and Management

Mechatronics

SE-100 44 STOCKHOLM

KTH Industriell teknik och management

Examensarbete MMK 2014:60 MDA 471

Visualisering för att underlätta modularisering av produktfamiljer

Fallstudie vid Hardware Radio Design, Ericsson AB

Elias Josefsson

Godkänt	Examinator	Handledare
2014-09-23	Martin Törngren	De-Jiu Chen
	Uppdragsgivare	Kontaktperson
	Ericsson AB	Kristian Lindskog

Sammanfattning

Syftet med denna uppsats är att undersöka hur visualisering kan hjälpa Hardware Radio Design på Ericsson AB i sitt modularitetsarbete. För att kunna tillmötesgå marknades krav på fler och fler versioner av radioenheter, har Ericsson infört en modulbaserad utveckling av sin produktfamilj. Med en modulär produktfamilj kan fler varianter utvecklas med samma mängd resurser och därmed stärka Ericssons position på marknaden. I arbetet med att fastslå moduläriteten måste olika intressenter förstå och utvärdera effekten utifrån flera aspekter.

Inledningsvis intervjuades utvalda intressenter på Ericsson för att identifiera den information, relaterad till modularitet, som gagnas av visualisering där utvecklingstid, resursanvändning och tidsplanering har fokuserats. Därefter utvecklades visualiseringskoncept för valda fokusområden som utvärderades av intressenterna. Koncepten innehåller kända diagram, t.ex. organisationschema för produktstruktur, stapeldiagram för resursanvändning och GANTT-schema för tidsplanering. Slutligen byggdes en web-applikation med flera interaktiva vyer och modeller för att kunna validera resultaten.

Slutsatsen i projektet är att en visualisering av produktfamiljens modularitet mycket sannolikt skulle hjälpa intressenter att förstå hur ett modulärt förslag påverka olika aspekter. Genom att, utifrån samma data, generera flera vyer kan aktörer med olika intressen tillsammans diskutera och fatta bättre beslut. Resultaten visar att ett visualiseringsverktyg för modularitet med stor sannolikhet också skulle underlätta kommunikationen mellan olika delar av organisationen.

Som framtida arbete bör en omfattande verifiering av visualiseringarna och verktyget göras för att säkerställa dess värde i en verklig utvecklingsprocess. Det finns även stora möjligheter att utveckla verktyget med fler typer av element, t.ex. mjukvara och testfall. Även integration med andra system kan ge ytterligare värden, t.ex. tidrapporteringssystem för att utvärdera gjorda tidsestimat med verkliga utfall.

KTH Industrial Engineering and Management

Master of Science Thesis MMK 2014:60 MDA 471

Multi-viewed Visualization of Modularity for Product Line Management

Case study at Hardware Radio Design, Ericsson AB

Elias Josefsson

Approved	Examiner	Supervisor
2014-09-23	Martin Törngren	De-Jiu Chen
	Commissioner	Contact person
	Ericsson AB	Kristian Lindskog

Abstract

The purpose of this thesis is to investigate how visualization may help Hardware Radio Design at Ericsson AB in their modularity work. As the market requires more and more versions of Radio Units, Ericsson has introduced a modular approach to product family development. With a good modularity, more variants can be developed with the same amount of resources and increase the profit margin. In the work of selecting modular plan, different stakeholders need to understand and evaluate the effect on multiple aspects.

Firstly, stakeholders at Ericsson were interviewed to identify what information related to modularity that would benefit from being visualized where development time, resource usage and time-plan have been focused. Secondly, concept designs of visualizations were iteratively developed for the focused areas. The concepts are built on well-known charts, e.g. organizational charts for product structure, bar charts for resource usage and GANTT chart for time plan. Lastly, a web-application was developed with multiple interactive views and models to be able to validate the results with stakeholders.

The conclusion of the thesis is that visualization of product family modularity very likely would help stakeholders to understand how modular plans affect different aspects. By supporting multiple views based on the same data, stakeholders with different viewpoints can together discuss and make better decisions. The results also show that it is very likely that a visualization tool for modularity would facilitate the communication amongst the whole organization.

As future work, a more extensive verification of the visualizations and tool are necessary to ensure its value in a real development process. Great opportunities exist for expanding the tool with more types of elements, such as software and test cases. Integration with other systems may also provide additional values, such as integration with time reporting system to evaluate time estimates with real outcome.

FOREWORD

This thesis is the last part of my master studies in Mechatronics at the Royal Institute of Technology (KTH) in Stockholm, Sweden. The project was conducted at Hardware Radio Design at Ericsson AB in Kista, Sweden.

Thanks go out to the following people, without whom this thesis would not have been possible:

Kristian Lindskog, supervisor at Ericsson, for all your help and support throughout the whole project. Your knowledge and experience have both inspired and guided me a lot.

DeJiu Chen, supervisor at KTH, for your feedback and guidance within all parts along the project.

Kristina Gold, manager at Ericsson, for giving me the opportunity to conduct this thesis at Ericsson and *Peter Eriksson* for the initial contact.

Mathias Holm, developing back-end of the tool as a separate thesis*,* for great cooperation and your open mind to all my ideas and requests.

Last but not least, I would like to thank *my family* for all your love, support and understanding throughout this period.

Elias Josefsson
Kista, June 2014

Notations

Symbol	Description
$L_i(t)$	Total workload coming from element i at time t
$L_{r,i}(t)$	Workload for resource r coming from element i at time t
$R_{log,num}$	Numerical reuse for logical elements
$R_{log,time}$	Development time reuse for logical element
$R_{phy,num}$	Numerical reuse for physical elements
$R_{phy,time}$	Development time reuse for physical elements
$R_{sub,num}$	Numerical reuse for subassemblies
$R_{sub,time}$	Development time reuse for subassemblies
T_i	Total development time for element i
$T_{r,i}$	Resource usage of resource r for element i
$T_{variant,i}$	Development time for element i per Frequency Variant
δ_i	Degree of coupling

Abbreviations

BOM	*Bill of Material*
CI	*Commonality Index*
DCI	*Degree of Commonality Index*
DSM	*Design Structure Matrix*
DSML	*Domain-Specific Modeling Language*
FA	*Final assembly*
GSM	*Global System for Mobile Communication*
KPI	*Key Performance Indicator*
LTE	*Long Term Evolution*
PA	*Power Amplifier*
PFMP	*Product Family Master Plan*
PLM	*Product Lifecycle Management*

RBS	*Radio Base Station*
RF	*Radio Frequency*
RU	*Radio Unit*
RX	*Radio Frequency Receiver*
SMA	*Surface Mount Assemble*
TCCI	*Total Constant Commonality Index*
TX	*Radio Frequency Transmitter*
UML	*Unified Modeling Language*
WCDMA	*Wideband Code Division Multiple Access*

TABLE OF CONTENTS

The introduction chapter describes the background of the project, its purpose and delimitations together with the used methods.

1.1. Background

Ericsson AB, in Kista, works with research, development and delivery of complete mobile systems. This particular thesis was conducted at the department of Hardware Radio Design (further on referred to as *Ericsson*).

As mobile data communication becomes more and more popular the market require more and more different versions of the Radio Units (RU). The number of different frequency bands has increased in the last years when operators are seeking for new frequency spectrums to use. In order to meet the market demands for data communication new platforms has also been developed.

1.1.1. Radio Units

Radio Base Station (RBS) is one of the key components in the radio access network that transmit and receive data between the Core Network and the Mobile Devices see Figure 1.

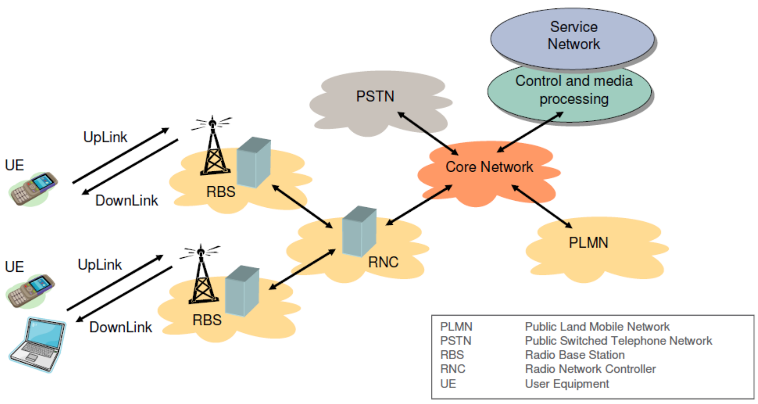


Figure 1. A generic Radio Access Network

It is the RU inside an RBS that handles all conversions of digital signals from the Core Network to an amplified analog radio signals, called downlink, and vice versa, called uplink. Figure 2 shows a general overview of the features in a RU.

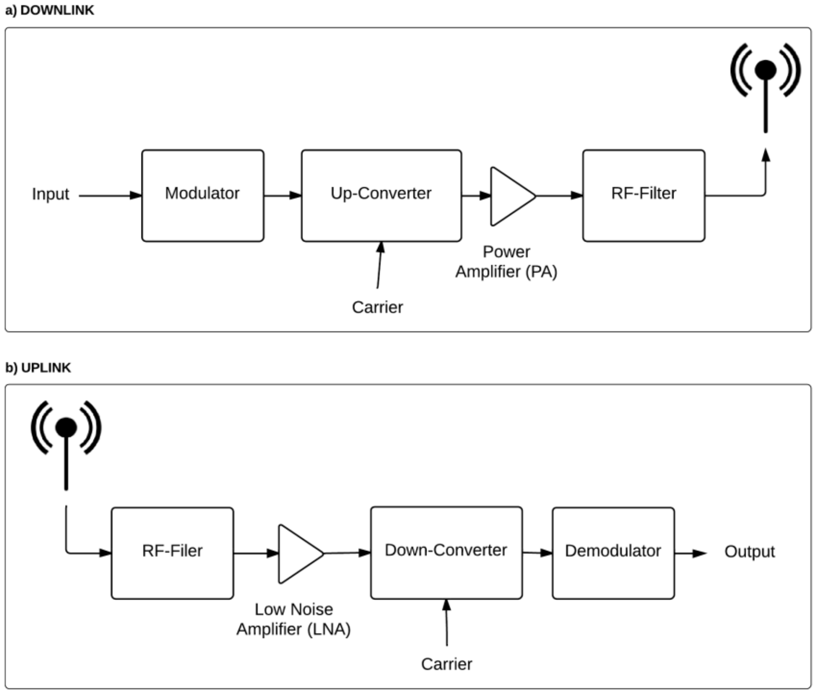


Figure 2. General overview of features in a RU,
a) Downlink; from Network to mobile unit, b) Uplink; from mobile unit to Network

On the downlink side (transmitter) the digital signal is modulated and up-converted for the given carrier. Before it is scattered by the antenna the signal is both amplified and filtered. On the uplink side (receiver) the signal, captured by the antenna, is filtered and amplified before it is down-converted, digitalized and demodulated to be sent back to the network. (Ohlsson, Norling, & Largren, 2013)

1.1.2. Variants of Radio Units

The growth of the mobile communication market has increased the need for different types of RUs. In urban environments, with high capacity demand, it is desired to have a high density of low-power units while less populated areas benefit from fewer high-power units (*Telecom Training*, 2007). Ericsson today has multiple types of RUs, see Figure 3. Two examples are the RUS that can be used in a Radio Base Station with multiple transmitters and receivers with a power of up to 60W/carrier and the DOT product that is used for indoor environments, connected and powered through existing copper wires.

Figure 3. The Product categories in Ericsson's Radio Unit portfolio

Some of the components in the radio unit are frequency specific, e.g. the filters. When the number of frequency bands increases the number of requested frequency variants also increases. On top of this the different standards (LTE, WCDMA, and GSM) also increases the demand of variety. Table 1 contains examples of variants developed for different products. The rows are different products with different performance, e.g. output power and standard, while the columns are different frequency bands. One check mark indicates thus one frequency variant of a product.

Table 1. Examples of products (rows) and frequency variants (check marks) for different frequency bands (columns).

	B0	B1	B2	B3	B4	B5	B7	B8	B9	B10	B12	B13	B14	B20
RRUS 01	✓	✓	✓	✓		✓		✓	✓				✓	✓
RRUS 11		✓	✓		✓	✓	✓				✓			✓
RRUS 02	✓			✓				✓						
RRUS 12		✓	✓	✓	✓	✓	✓	✓		✓		✓		
RRUS A2			✓	✓	✓		✓							

1.1.3. Challenges

To cope with the growing number of different RUs, Ericsson has, as many other companies, developed a modular approach to product family development. This gives the ability to reuse parts between Frequency Variants within a product family.

When modularizing the RUs it is important for all stakeholders to be able to see how different modular approaches affect their work and responsibility. Only then it is possible to find a good decomposition of the system and maximize the number of Frequency Variants within the given amount of resources.

With an increasing number of modules and Frequency Variants (about a couple of hundreds today) the need of a tool increases to collect, keep, maintain and not least visualize the modularity data within the multi-disciplinary development.

Previous work has been done at Ericsson to investigate available modularization visualization tools in the market, mostly optimized for the automotive industry, without satisfying results. Within the scope of a master thesis, Hesami (2012) also investigated the ability to use static and dynamic Design Structure Matrix (DSM) to solve the problem. Both approaches have given good insights, but the problem how to visualize modularity information from different perspectives is still not solved.

1.2. Motivation and Purpose

The overall purpose of this thesis is to help Ericsson in their work to find the best modularity of its product family. By using a modular approach Ericsson hopes to deliver more variants with the same resources and hence strengthen its competitiveness and profit.

Figure 4 illustrates how visualization plays a role in the work of modularity. Ericsson has decided to use a modular approach to successfully deliver competitive products to the market. To find the best modularity, previous work has investigated Key Performance Indicators (KPI) and the ability to use DSM to compare different modular approaches.

Ericsson has also identified a need to visualize the modularity where different approaches can be compared and make the discussions more effective. This is the major purpose of this thesis, to investigate how visualization can be used within the work of finding the best modularity and to communicate how the modularity will affect different aspects.

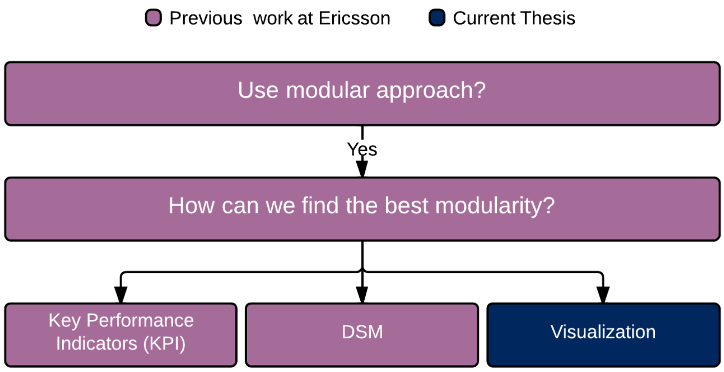


Figure 4. Illustration of this thesis related to Ericsson's modularity work.

1.3. Research Question

Four questions will be investigated within this thesis:

1. What modularity information would benefit visualization to help multiple stakeholders, with different viewpoints, to select the best modular plan?
2. What modularity information would benefit visualization to help the organization to understand the consequences of modularity?
3. What views and models are needed to visualize identified modularity information?
4. How can these views and models be realized in a tool?

1.4. Hypothesis

Two hypotheses are defined and tried to verify throughout the project.

- By visualizing the modularity, and its effect on selected aspects, stakeholders get a better decision support when choosing decomposition of their product family. A visualization of modularity also facilitates the communication within the organization about modularity and its effects on multiple aspects.

- The visualization can be realized in a web tool based on existing JavaScript visualization libraries and a relational database. Web-technologies are suitable since it has good support for great visuals and user-access without any extra plug-in installation.

1.5. Method

Three different methods were used throughout this project. Studies of Ericsson internal documents were used to get a knowledge base about Ericsson's organization and product family. Deep interviews were used to identify key stakeholders' viewpoints and needed views. And finally the visualization of views were iteratively developed and validated with stakeholders. First by sketches made by pen and paper and later building a proof of concept with existing front-end web technologies.

1.6. Delimitations

This work has identified key stakeholders, with different viewpoints, that have an interest in modularity information. Based on stakeholder's viewpoints, multiple views were identified. To complement previous work at Ericsson, views related to organization and process were selected for deeper analysis:

- Development time
- Time-to-market
- Resource workload

Based on identified views, investigation was done to see how the views best could be modeled and visualized. A proposal how to structure the modularity data for generating visualizations was also developed and a proof of concept was built with existing web technologies. A back-end was developed by Holm (2014) in another thesis. The whole process is visualized in Figure 5.

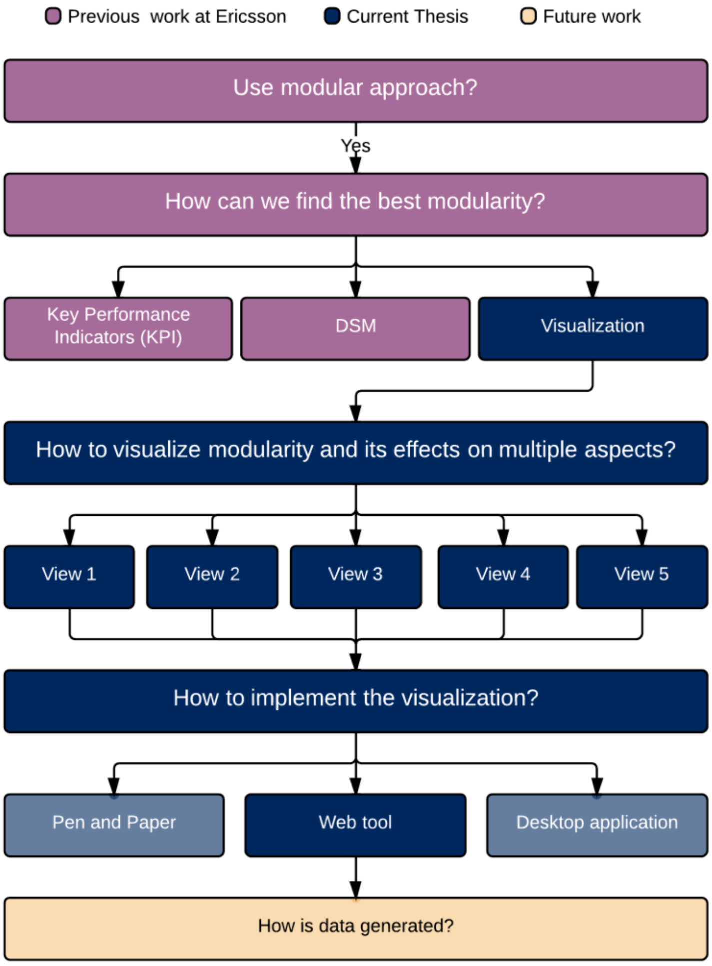


Figure 5. Illustration of current thesis related to Ericsson's modularity work and future work.

2. FRAME OF REFERENCE

This section contains a summary of relevant research related to modularity, strategies and drivers of modularity and some examples of methods and tools used in modular design.

2.1. Modularity for product-lines

When the variety of products increases, due to customer desire for customization on a competitive market, companies have been forced to find new ways to deliver more products with shorter time-to-market and same amount of resources. One of the more popular solutions is modularity where products are designed by combining modules shared amongst multiple products. (Jiao, Simpson, & Siddique, 2007)

2.1.1. Basic Concept

Jose and Tollenaere (2005) define modularization as *"an approach to organize complex designs and process operations more efficiently by decomposing complex systems into simpler portion"*.

When having a modular design it is possible to let multiple variant use the same modules. Figure 6 contains a simple example where different combinations of five unique modules can result in four unique Frequency Variants. If all modules in every Frequency Variant had been unique, it would require twelve individual modules.

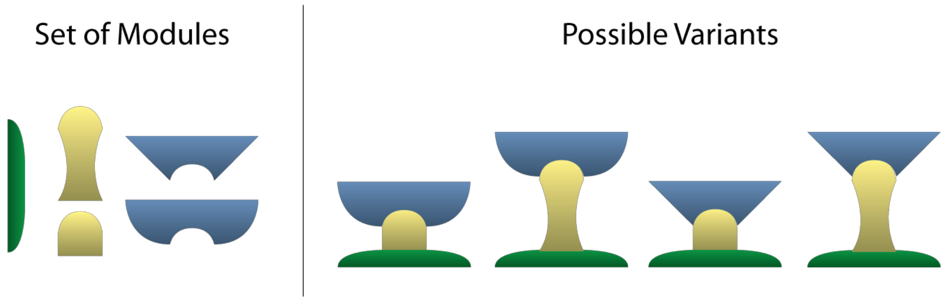


Figure 6. Example of modular design

When designing a product line it is important to find a good balance between the number of common modules and specialized modules. The use of more specialized modules increases the number of possible combinations, but increases also the cost. The goal is to find a maximum number of standard modules without affecting the ability to develop the necessary products. (Jose & Tollenaere, 2005)

Robertson and Ulrich (1998) describe it very well with an example of two products. If they would share 100% of the modules they would be identical and the distinctiveness would be zero. But, on the other hand, to reach a total distinctiveness they would need to have totally individual modules. The introduction of unique components does not automatically give a higher distinctiveness. Figure 7 shows the trade-off between distinctiveness and common parts, where Architecture 3 is the best in aspects of modularization.

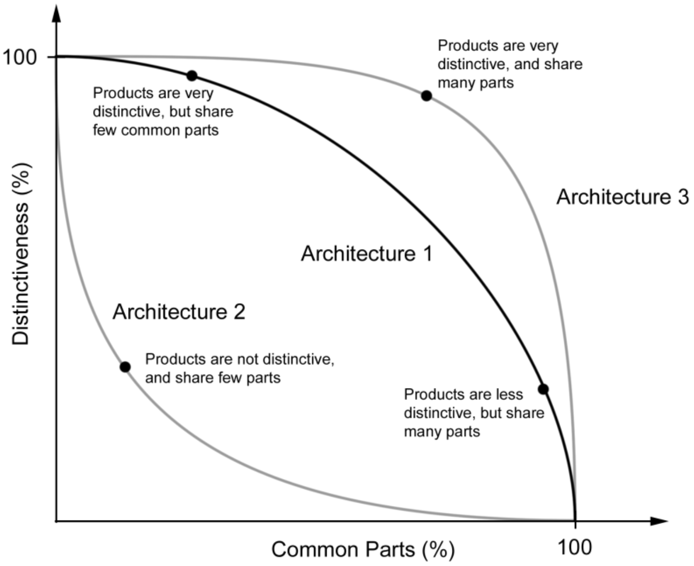


Figure 7. The trade-off between distinctiveness and commonality.

2.1.2. Strategy of modularity

Even if there is a big diversity amongst different methods working with modularity, Liu, Wong, and Lee (2009) have identified two general strategies; top-down and bottom-up. Top-down, also called proactive approach, focus on the platform developing a set of modules that creates new products, see Platform B in Figure 8. Bottom-up, or reactive redesign, starts with existing products that are redesigned to be more standardized with a modular approach, Platform A in Figure 8. The strategy to use depends on the current situation for a company. If there is a new market opportunity without a clear defined customer desire, top-down strategy can be used to enable big flexibility of products, within same platform, to meet new customer needs. One example is Sony that developed one technical platform to create hundreds of Frequency Variants in its Walkman line to meet the quick changes in customer requirements and new technology.

If a set of products already exists with clear market potentials, bottom-up modularity strategy can be used to streamline the development, manufacturing, supply and maintenance to secure the quality and reduce the cost. Black and Decker, for example, did rebuild their whole product line around a common motor platform.

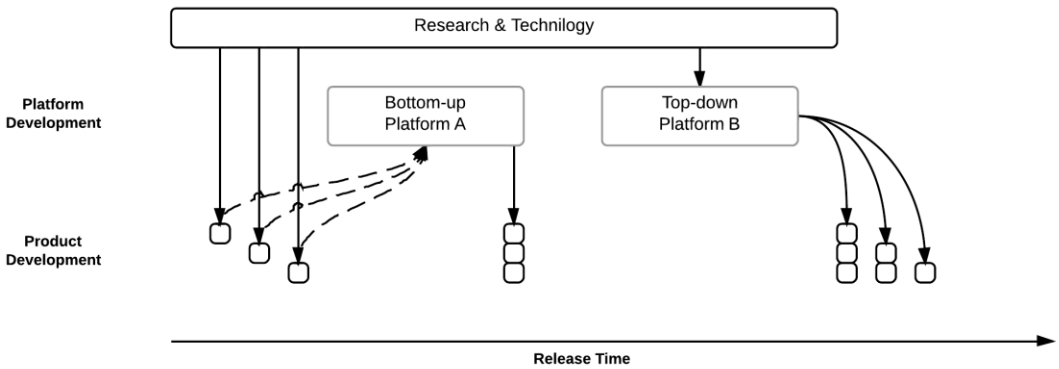


Figure 8. Bottom-up and top-down strategy for platform design.

2.1.3. Drivers of modularity

A product will go through multiple stages during its lifetime; starting with a customer need through development, manufacturing, service and finally end with recycle. Each step corresponds to the voice of a stakeholder that has different requirements for the product, see Figure 9.

Figure 9. Voices of interests during a product lifetime

Each "Voice of X" will affect the strategy of modularity. Lange and Imsdahl (2014) have mapped the voices to module drivers, which are presented and described in Table 2. Each module driver might have different reasons for modularity based on their viewpoint.

Table 2. Corresponding module drivers to Voices of X

Voice of	**Module Driver**
Customer	***Different Specification*** – impacts the need for different technical performance between products due to variance in customer needs
	Styling – impacts the need of variance in product appearance to meet different customer groups
Engineering	***Carry Over*** – impacts the reuse of modules across products and product generations
	Technical Evolution – impacts the development strategy to be able to implement new technology coming from external forces.
	Planned Design Change – impacts internal strategy to launch new products to meet new customer needs or reduce the cost.
Manufacturing	***Common unit*** – impacts the strategy to reduce the number of different physical forms for a specific function.
	Process and/or Organization – impacts the strategy to make sure there are available efficient processes for manufacturing modules/products.
Quality	***Separate Testing*** – impacts the strategy to make functions testable independently of the products.
Supply Chain	***Supplier Availability*** – impacts the strategy to use "black box" technology that can be outsourced to suppliers.
After Market	***Service and Maintenance*** – impacts the strategy of service and support of a product.
	Upgrading – impacts strategy to be able to upgrade existing product to extend product life.
	Recycling – impacts the strategy to simplify the recycling, e.g. usage of homogenous material and codes regarding the disposal.

Some of the module drivers are coinciding, while other might be conflicting. An example of a conflict is the Technical Evolution and the Carry Over, where Technical Evolution wants to make changes to a module due to new technology, while Carry Over wants to reuse the same module without any changes in a new version of the product.

How much impact each voice will have on the modular design depends on the company's business strategy. The business strategy should answer the question of how the company will add value to the customer, the Value Discipline. Figure 10 shows the three generalized value disciplines; Product Leadership, Operational Excellence and Customer Intimacy. The choice of discipline will affect the priority of the different voices.

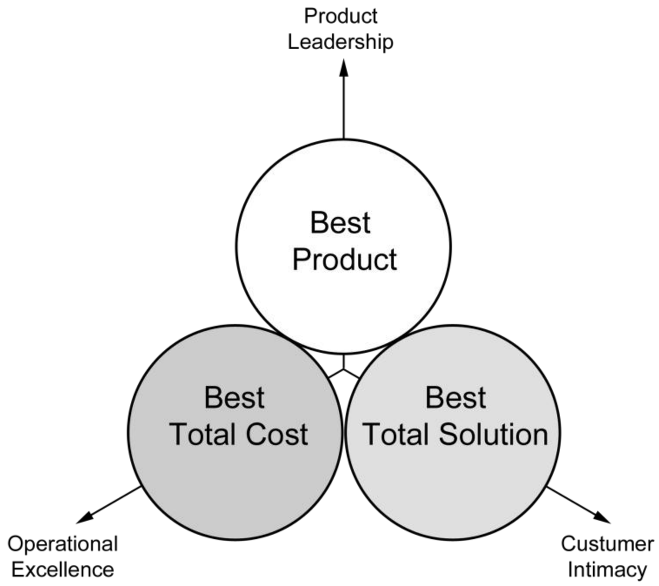


Figure 10. An abstract representation of the value disciplines.

Product Leadership companies want to deliver the very best product for the customer and prioritize the Technical Evolution over Reuse and Common Units. Customer Intimacy companies try to deliver a customized product to meet a single customer and therefore prioritize Different Specifications, Styling, Service and Maintenance with focus on the total solution. Operational Excellence companies are focused on the price and need to prioritize Process and/or Organization, Carry Overs and Separate testability to keep the cost down.

To be able to find the best modularity for a product line, it is important to know where a company's business strategy is located in the Value Discipline Space. If not, it will be very hard to weigh between the different drivers of modularity. Lange and Imsdahl (2014) recommend companies to choose one primary value and eventually one additional value to make the strategy work easier.

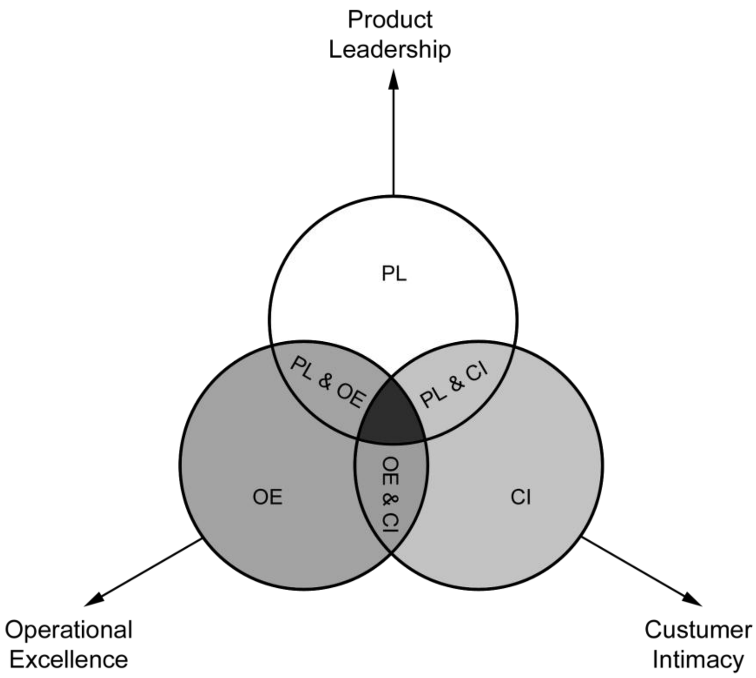


Figure 11. Companies are recommended to select a position outside the central area to simplify the weigh between drivers of modularity.

2.2. Modular Design Methods

When the importance of modular design has increased for companies, multiple methods and tools have been developed to support the design process. The goal is always to find the optimal modularization, but since the definition of *"optimal"* varies a lot, multiple methods have been developed.

This section will only look briefly on some types of methods that are relevant to this thesis. For the interested reader, there are literature reviews done on a wider range of methods. (Fixson, 2003; Jiao et al., 2007; Jose & Tollenaere, 2005; Simpson, 2004)

2.2.1. Grouping or Clustering

The goal is to group elements in clusters so that interactions within clusters are high, but interactions between clusters are low. For this DMS matrixes are advantageously used, where elements can be functional or physical elements. Within the matrix, the dependencies are specified between elements and algorithms can cluster elements automatically. Eppinger (1994) suggests using four variants of dependencies; Spatial, Energy, Information and Materials. Hesami (2012) uses clustering to identify what functions in the RU to group in module, based on the existence of interfaces between functions, see Figure 12.

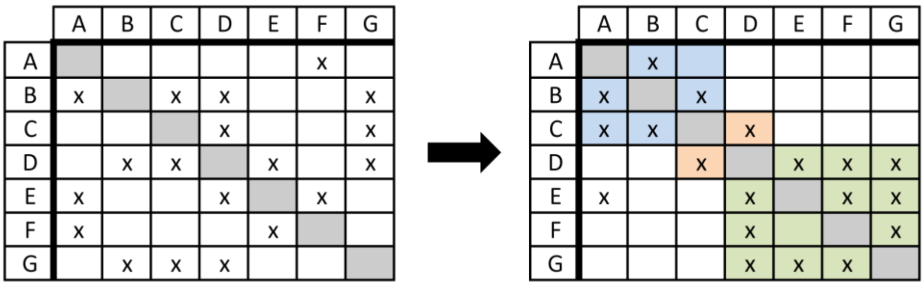


Figure 12. An example of clustering using DSM.

2.2.2. Figures of merit

By creating figure of merits related to modularity, it is possible to evaluate different modularity options. There are multiple indices developed by researchers where some examples, used by Ericsson to evaluate their RUs, will be described.

Degree of coupling

A component highly dependent on interfaces with other components has a high degree of coupling. The microprocessor in a PC is one example of a component with a high degree of coupling since it has approximately 200 interfaces to other components. Mikkola and Gassmann (2003) defines the degree of coupling for a whole system as

$$\delta_i = \frac{\text{total number of interfaces in subsystem } i}{\text{number of components in subsystem}} = \frac{\sum k_c}{n_c}. \quad (1)$$

A system or sub-system with a high degree of coupling may be hard to decompose and the indices can help to have feasible expectations when looking at other figures of merits.

Degree of Commonality Index, DCI

The Degree of Commonality Index (DCI) reflects the average number of common parent items per average distinct component part (Thevenot & Simpson, 2006). DCI is defined as

$$\mathrm{DCI} = \frac{\sum_{j=i+1}^{i+d} \Phi_j}{d} \tag{2}$$

where Φ_j is the number of immediate parents component *j* has over a set of variants, d the total number of distinct components in the set of variants and *i* the total number of end items or the total number of highest level parent items for the product structure level(s). An example is displayed in Figure 13.

Total Constant Commonality Index, TCCI

Total Constant Commonality Index is a modified version of DCI with absolute boundaries between zero (0) and one (1) and defined as

$$\mathrm{TCCI} = 1 - \frac{d-1}{\sum_{j=1}^{d} \Phi_j - 1} \tag{3}$$

The TCCI facilitates comparisons between different product families and within a product family during redesign. (Thevenot & Simpson, 2006). An example of calculation is displayed in Figure 13

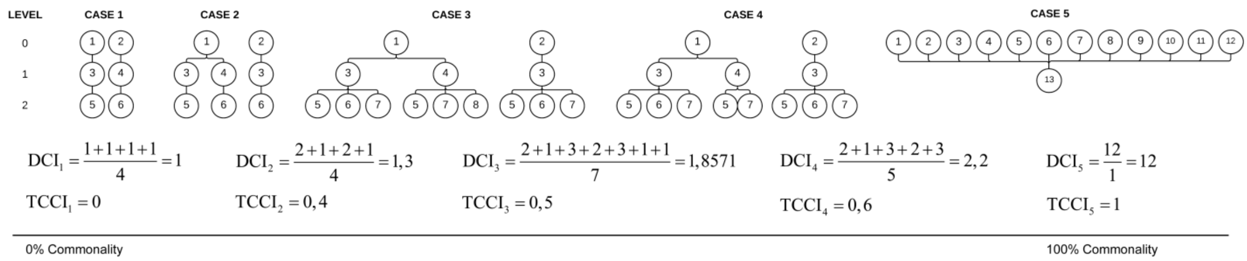


Figure 13. Example of calculation of DCI and TCCI(Wacker & Treleven, 1986)

Commonality Index, CI

Communality Index is also a modified version of DCI that indicate the level of unique parts:

$$\mathrm{CI} = 1 - \frac{u - \max p_j}{\sum_{j=1}^{v_n} p_j - \max p_j}, \tag{4}$$

where *u* is the number of unique parts, p_j is the number of parts in model j , and v_n is the final number of variants (Thevenot & Simpson, 2006).

Six variants of computer mice, each having 20 parts without any reuse would give

$$\mathrm{CI}_A = 1 - \frac{120-20}{120-20} = 0, \tag{5}$$

and if the number of total needed components were reduces to 70 it would give

$$\mathrm{CI}_B = 1 - \frac{70-20}{120-20} = 0.5. \tag{6}$$

2.2.3. Trade-off Optimization

In the work of finding the best modularity, multiple optimization methods are suggested by researchers. The objectives of the optimization are expressed in mathematical terms and solved with computer algorithms. Often the methods contain multi-constrain optimization with some basic assumptions. An example is mentioned by Simpson (2004):

1. When maximizing each product's performance the demand for the product is also maximized.
2. When maximizing commonality amongst products the production cost is minimized.
3. The optimal product family is found when resolving the trade-off between the previous two assumptions.

The number of stages differs between optimization methods. Single-stage methods seek to optimize the whole product family and its products in one stage, while two-stage methods start with optimizing the platform and the products in stage two. (Simpson, 2004)

The choice of programming approaches varies a lot and it seems like the method nature decides what programming approach is chosen. Some examples are *sequential linear programming*, *sequential quadratic programming*, *nonlinear programming* and *generalized reduced gradient*.

The challenge with mathematical optimization methods is to simplify and translate the real problem to a mathematical description. The problem is not to get the right answer, but ask the right question.

2.3. *Visualization*

The saying “A picture is worth a thousand words” is well known and something most people can confirm. This section contains research regarding the importance of visualization in general and some examples of visualization tools used for modularity.

2.3.1. Importance of Visualization

Visual graphics are very useful in the process of making information understandable and accessible. Tufte (1983) describes the importance of visual display when working with quantitative information, both for own analysis and presentations for an audience. As a simple example, four small data sets are presented in Table 3 where the patterns are hard to see.

Table 3. Four different set of data hard to analysis without any visualization.

1		2		3		4	
X	y	x	Y	x	y	X	y
10,0	8,04	10,0	9,14	10,0	7,46	8,0	6,58
8,0	6,95	8,0	8,14	8,0	6,77	8,0	5,76
13,0	7,58	13,0	8,74	13,0	12,74	8,0	7,71
9,0	8,81	9,0	8,77	9,0	7,11	8,0	8,84
11,0	8,33	11,0	9,26	11,0	7,81	8,0	8,47
14,0	9,96	14,0	8,10	14,0	8,84	8,0	7,04
6,0	7,24	6,0	6,13	6,0	6,08	8,0	5,25
4,0	4,26	4,0	3,10	4,0	5,39	19,0	12,50
12,0	10,84	12,0	9,13	12,0	8,15	8,0	5,56
7,0	4,82	7,0	7,26	7,0	6,42	8,0	7,91
5,0	5,68	5,0	4,74	5,0	5,73	8,0	6,89

When plotting the data in diagrams, the characteristic of each data set is revealed very well and the underlying information of the data is much easier to understand, see Figure 14.

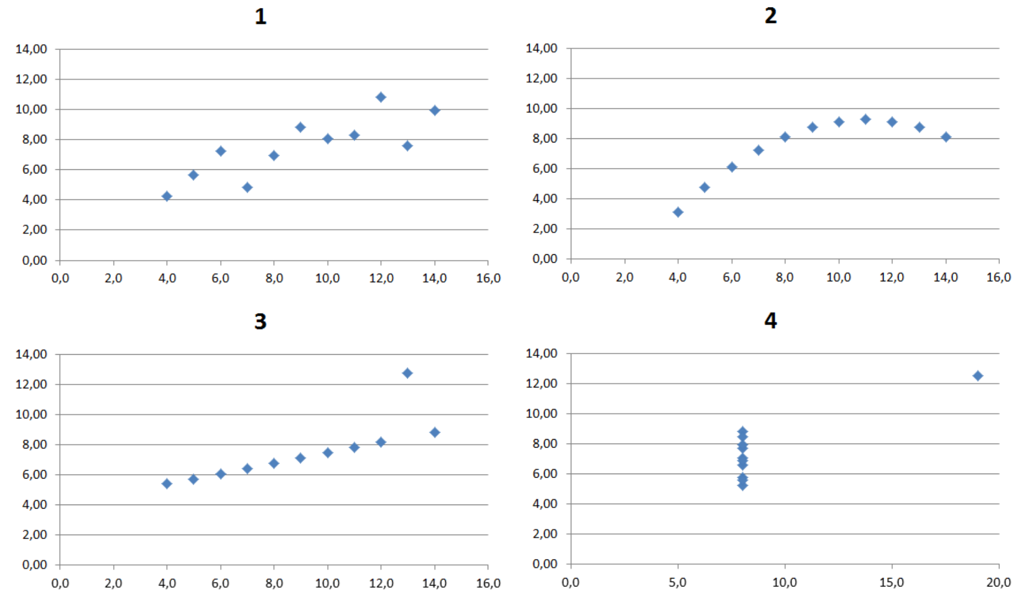


Figure 14. The visual presentation of data-sets presented in Table 3.

Visuals are not only useful when working with a big amount of data, but also to compare different configurations. Yau (2013) describes the Coffee Drink Illustration, created by Lokesh Dhakar, shown in Figure 15, as a great example of a visual description of ingredients for different coffee drinks. By using the context of a coffee cup, the illustration shows both the relative amount of ingredients per variant, but also the relative total amount between variants.

Figure 15. Coffee Drinks Illustrated by Dhakar (2007)

Visual representations are also an important tool when making decisions. Tufte (1997) describes two examples; one good visual representations leading to great decisions, and one bad visual representation leading to catastrophe.

The first example is how John Snow could identify a street pump as the cause of the Cholera Epidemic in London in 1854. He marked out all deaths from cholera (▮▮▮▮) on a map together with all the community water pump-wells, see Figure 16. With the map visualization, he could see the general pattern: High density of deaths around the pump at Broad Street. But there were some buildings in the neighborhood that had a very low rate of death that Snow did investigate further. A brewery, located close to the poisoned pump, had a very low death-rate. The reason was that the brewery gave the workers a certain amount of malt liquor and very few workers drank anything else. By visualizing the data related to the epidemic, Snow could convince the people in charge to remove the handle of the poisoned pump and the epidemic was stopped.

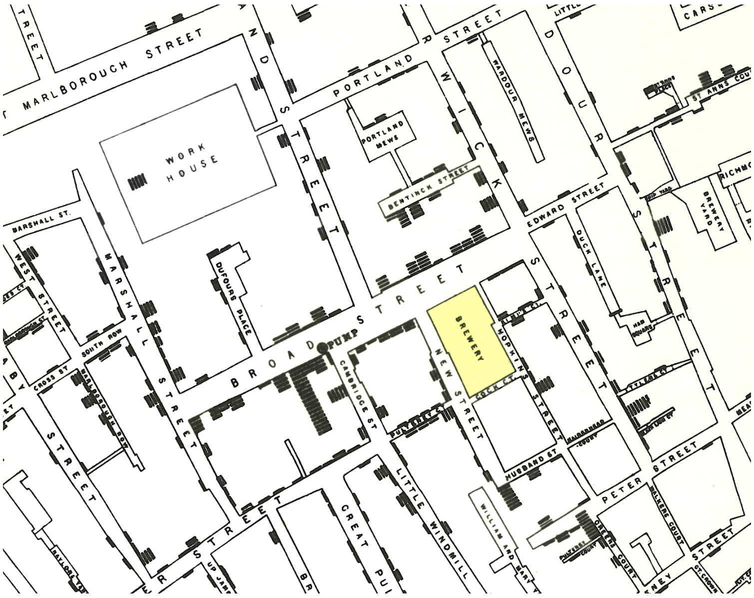


Figure 16. The deaths from cholera were indicated on a map () to reason and argue about the cause of the epidemic. (Snow, 1855)

The other example is when the space shuttle Challenger exploded in 1986 where seven astronauts died. The night before the launch, engineers tried to convince the managers that some O-rings could sever big problems due to the cold weather. Engineers prepared 13 charts with evidence to convince NASA to cancel the launch. A combination of cultural differences, bureaucracy and bad visual argumentation hindered the engineers to stop the launch. Tufte shows how the data behind the 13 charts are correct and probably enough to convince the managers to cancel the launch if it was presented and visualized in a clear and convincing way.

2.3.2. Visualization of modularity

When developing complex products, each sub-system is often developed by an individual group with experts in that specific area. The engineers within the groups have a good understanding of their own system and often also made an own visualization of their system. There is also always a need of sharing information between groups to ensure smooth integration, and the need increases even more when using a modular product design. It is important that everyone knows how a module will interface with multiple variants of other modules and how a change in a module will affect the product line. A visualization of the whole complex system is a great tool to help everyone get a good overview of the system. (Bruun, Mortensen, & Harlou, 2013)

The challenge, when trying to visualize a complex system, is to find a good balance between the details and the amount of information. It's very hard to visualize a system in only one view without losing the depth in the information. Therefor multiple views and models are needed to grasp the whole complex system.

2.3.3. Examples of visualization used in modular design

There are multiple approaches how to model and visualize modularity. The DSM described previously is in one sense a visualization, where the dependencies are shown between elements. Unified Modeling Language (UML) is also used to visualize the structure of products and possible modules (Jose & Tollenaere, 2005). With a unified language, syntax and semantics are standardized and give a good common understanding amongst stakeholders, without any need of specific guidance. But to be able to model the domain specific aspects of a system, Domain-Specific Modeling Language (DSML) is used where the syntax and semantics are defined specifically for a domain.

Harlou (2006) suggests two models to visualize variability within product families; *Generic Organ Diagram* and *Product Family Master Plan (PDMP)*.

Generic organ diagram is based on a block diagram that is typically used within modelling architectures. The blocks represent organs that can be related with different interfaces. If there are variants of an organ, these will be illustrated as multiple blocks stacked upon each other, see Figure 17. One example is a computer where the hard drive might have different storage capacity.

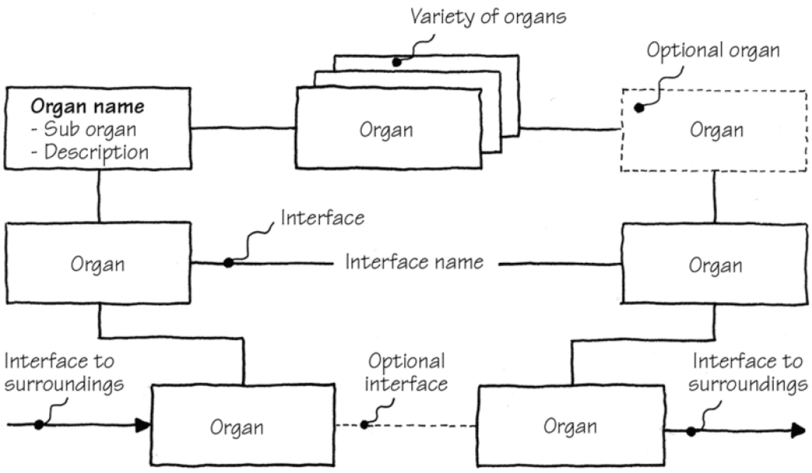


Figure 17. The basic components of a Generic Organ Diagram. (Harlou, 2006)

The Generic Organ Diagram can combine multiple products to illustrate both shared and individual organs.

PDMP is inspired by object-oriented formalism often used in software development, with Objects, Relations, Attributes and Instances. Three types of relations are used with different meanings, see Figure 18.

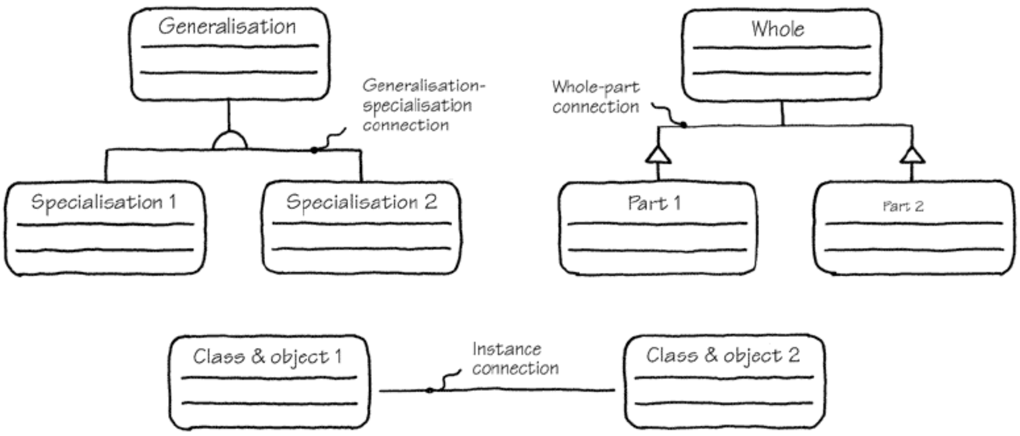


Figure 18. The three different types of relation in *Product Family Master Plan*. (Harlou, 2006)

To illustrate how it is used an example of a car family is presented in Figure 19. Starting from the top left, a Car can be of one of the specializations; Sedan, Station wagon or Van. A car contains one engine (part) that can be one of the three specializations (1.6, 1.8 or 2.0 liters). The engine also contains multiple parts, e.g. Engine block and Piston that comes in different specializations. This structure continues for all parts in the car and describes how different car models are built up by shared and individual specializations of parts.

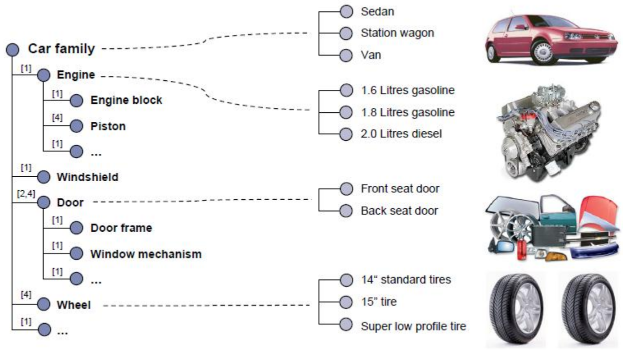


Figure 19. An example of a car family modelled with PFMP. (Harlou, 2006)

PFMP includes three views to cover different aspects of the product family:

- *Customer view* – Describes the product family from a customer's point of view and should answer questions like: "What are the features and application and how does it add value to the customer?"
- *Engineering view* – Describes the product from an engineering point of view and describes the structure of containing organs. Should answer questions like "How does the product family work?" and "How do the organs varies?"
- *Part view* – Describes the physical entities of the product family. Should answer questions like "How is the product family realized physically and how does it vary?" (Figure 19)

Bruun et al. (2013) argue that many methods lack the ability to create models that support the whole design process from concept to detailed design. Bruun et al. divide methods into two groups: (a) methods that are visual but have the aspects related to a single or few requirements in focus (Functional structures, PFMP, Generic Organ Diagram) and (b) methods with broader focus but with a lack of visual quality (DSM, Generic BOM). Therefor Bruun et al. introduce the Interface Diagram to combine good visual quality and broader focus. Interface Diagrams, see Figure 20, are static visuals that can be stored within existing solutions for Product Lifecycle Management (PLM).

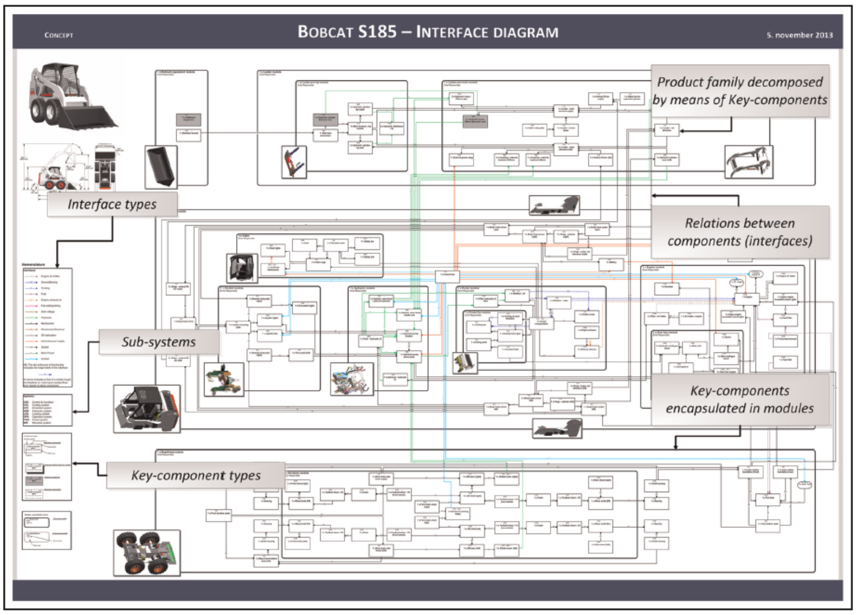


Figure 20. An example of Interface Diagram. (Bruun et al., 2013)

2.3.4. Computer Tools for Visualization

When the amount of data grows, the usage of a computer tool to store and structure data increases. There are multiple available visualization tools covering different aspects of the work with modular product design.

MetaEdit+

MetaEdit+ is a tool for Domain Specific Modelling where the workflow is divided into two steps (Metacase, 2014). In the first step a meta-model is defined containing the language rules, notations and generators. In the second step multiple models are designed based on the meta-model, see Figure 21. By defining a meta-model for a product family, MetaEdit+ can be used to visualize the structure and relations between products and elements. But MetaEdit+ does not contain native features for displaying general charts as pie-chart, area-chart and bar-chart. This has to be done by exporting data and visualize in other tools.

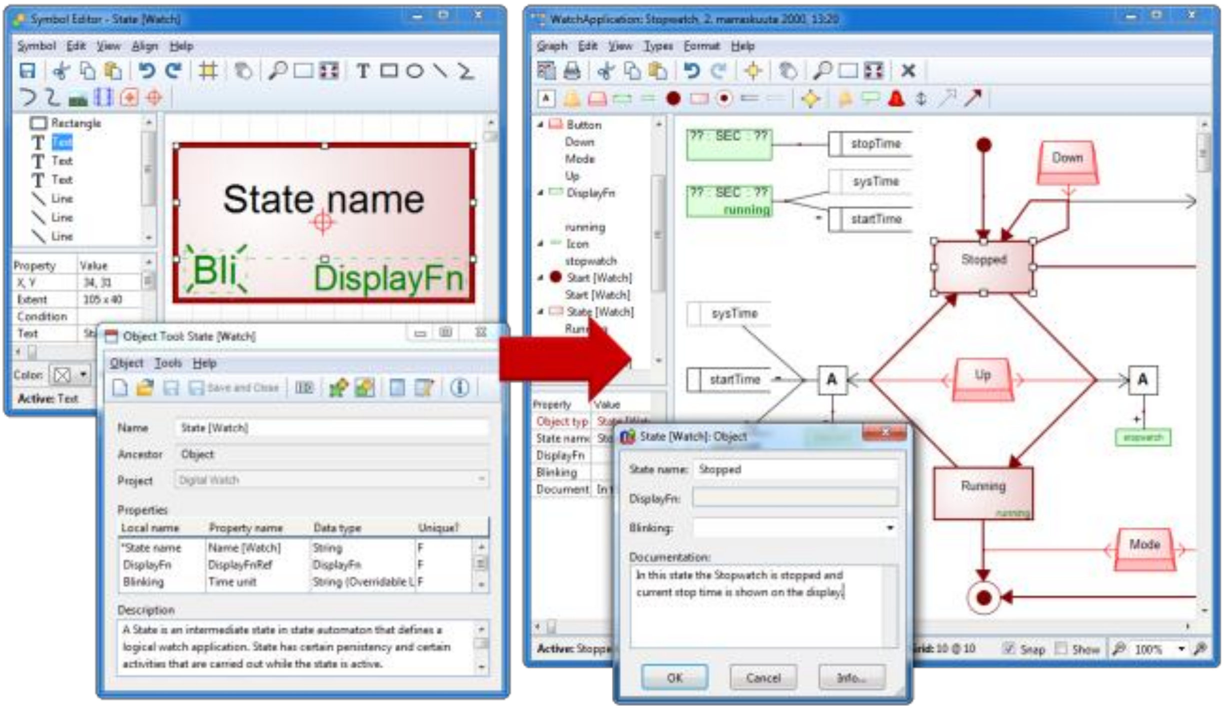


Figure 21. Screenshots of MetaEdit+, with its two steps; meta-model definition and modelling. (Metacase, 2014)

Birst

Birst is a web-based Business Intelligence System for visualizing and analyzing data (Birst, 2014). Multiple data sources can be imported and modelled to generate views and reports for desired purpose, see Figure 22. The tool enables big amount of data to be visualized in multiple types of charts. But since it is targeting business analysis it does not support any visualization for product structure and dependencies.

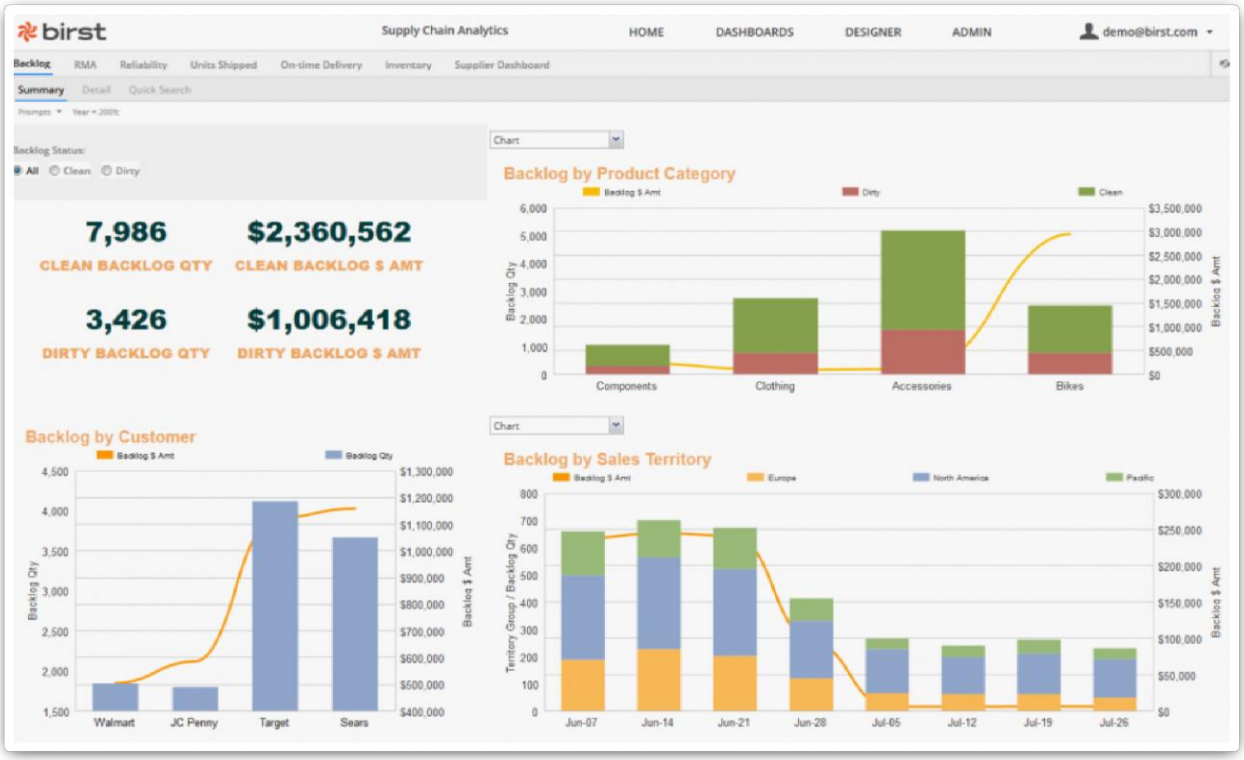


Figure 22. A screenshot of Birst, a Business Intelligence System. (Birst, 2014)

Power View

Power View is a part of Microsoft Office 365 enabling interactive visualization from a big set of data, see Figure 23 (Microsoft, 2014b). The features are similar to Birst and also lack features to visualize product structure and dependencies.

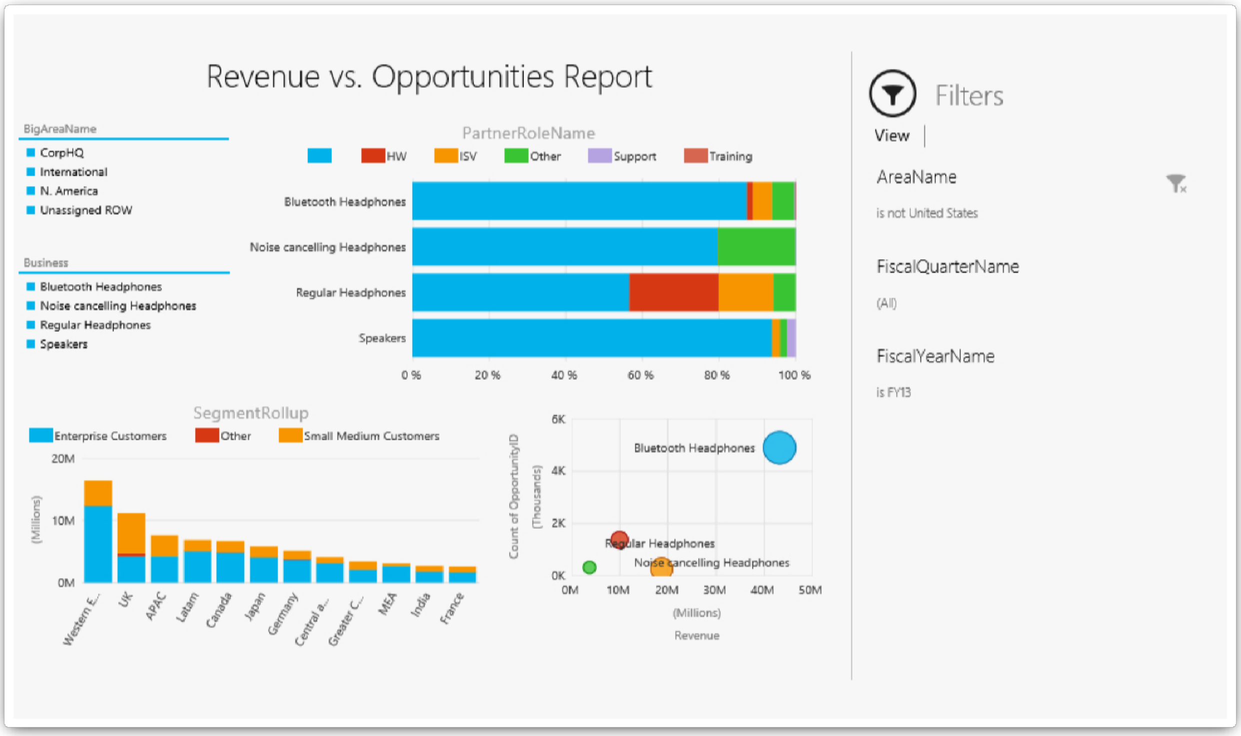


Figure 23. Screenshot of Power View, a visualization tool for Microsoft Office 365.

2.3.5. Frameworks for developing customized tool

If existing tools do not meet all requirements, tools can be built for specific purpose by using existing frameworks. Three examples, that all have been used within Ericsson, are presented.

Microsoft Office

Microsoft Office is a well-used and well-known product family within Ericsson and used for multiple purposes (Microsoft, 2014a). Rotor is a tool developed at Ericsson for estimating manufacturing cost with great results. The tool is built by combining Excel-files, Access database and Visual Basic macros.

There are some built-in features for data visualizing in Microsoft Office, e.g. pie-chart, area-chart and bar-chart, while other visuals can be developed by using Visual Basic and shapes.

Desktop Application

A desktop application is installed and runs stand-alone on a desktop or laptop computer. There are multiple languages available for development, e.g. Java, C++ and Python. Java has previously been used within Ericsson to develop tools and contains a lot of libraries for visualization (Fekete, 2004; "Swing Depot: Component Suites," 2004). One example is yFiles, shown in Figure 24, supporting multiple ways to visualize structure and dependencies ("yFiles for Java," 2014).

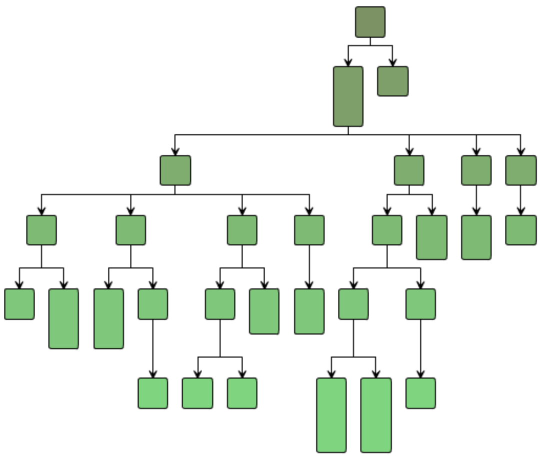


Figure 24. Organizational chart is one of the ways to visualize structure and dependencies in yFiles. ("yFiles for Java," 2014)

Web-application

A web-application is a software application that runs in a web-browser. The usage of web-application has increased over the last years ("Survey: Firms Increasing Reliance on Web Application Security Testing Tools," 2013). Fowler and Stanwick (2004) address the advantages for using a web-applications compared to a traditional desktop application.

1. Web-applications don’t have to be compiled to run
2. Browser-based applications are portable, even mobile.
3. Applications delivered through a browser do not have to be installed on individual computers.
4. The web is visual and supports great abilities for a better user experience.

Fowler and Stanwick also reveal the disadvantages of using a web-application.

- Browser incompatibilities might lead to different experiences depending on user’s browser.
- Performance can be slow.

By building an own tool based on existing libraries, it is possible to customize the tool for a target group without needing to develop everything from scratch. There are multiple available JavaScript visualization libraries available for free or very low cost. Some examples are D3.js (Bostock, 2013), Basic Primitives ("JavaScript/HTML Organizational, Family, PERT & Dependencies Chart," 2014) and ChartJS ("ChartJS - HTML5 JavaScript dynamic client side data visualization," 2014) that all support different type of high quality visualization.

This section describes the iterative process where key stakeholders within Ericsson have been interviewed to identify what information needed and how to visualize it. In total five people have been interviewed, two working within pre-development, one working with supply and two working with development planning. Based on the interviews a prototype of a visualization tool have been developed and verified with the stakeholders.

3.1. Identifications of Stakeholders and their Viewpoints

The pre-development team at Ericsson is responsible for securing that technology is ready for usage in new products. They are also responsible for the planning of platforms and how the products ordered from product management are going to be implemented.

As previously described, the modular approach will be a core strategy to fulfil the requirement of increasing number of variants with the same resources. There are multiple stakeholders throughout the product lifetime that will be affected by the modularity, see Figure 25. The pre-development team tries to find the best modularity based on the input from the stakeholders.

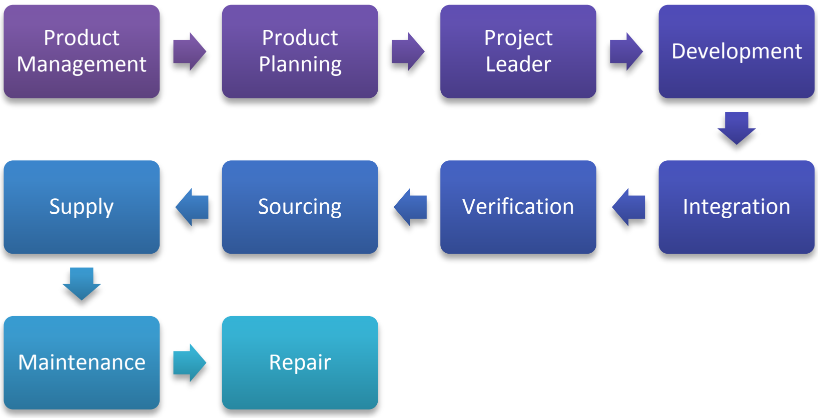


Figure 25. Stakeholders throughout the whole product life cycle

Ericsson previous work related to modularity optimization has been focusing on the actual product and its interfaces and dependencies. During the interviews, other challenges relating the organization and process have been discovered. Product performance is obviously very important, but to select the best modularity it is also important to understand how it will affect the organization and the development process. To complement previous work, the focus is to investigate and visualize aspects related to organization and process.

The rest of this section presents all identified viewpoints and their aspects of modularity.

3.1.1. Product management

Product management identifies market needs and decides how Ericsson will bring value to the customers, see section 2.1.3. This includes specifying what products that needs to be developed and their overall requirements. The customers' needs are not only related to product performance and multiple frequencies, but also the customers' different business strategies. Some customers want the latest technology and are willing to pay for it while others are price sensitive and willing to sacrifice some performance for a lower price. Therefor product management wants to deliver multiple variants of products to fulfil different customer requirements for a reasonable price. With a good understanding of modularity and how it will affect the product family, product management could increase their ability to request products that fits the platform and still meets the variety in customers demand.

3.1.2. Product Planning

Product Planning creates a plan for the development of requested products from Product Management. This includes release plans for multiple products and variants together with allocation of resources to ensure that products can be released on time. Most of the required products are well specified and Ericsson uses mainly a bottom-up approach, described in section 2.1.2, to streamline the development, manufacturing, supply, etc. to reduce the cost. But sometimes new products are requested, based on changes in the market. The top-down approach is therefor also applicable.

A modular approach with many shared modules will increase the resource demands and development time for the first variants, but decrease the same for later variants when all the common modules are developed. A visual representation of how different modular plans affects total development time, resource usage and release plans would be a great tool when analyzing the best modularity and to argue about it within the organization.

3.1.3. Project Leader

Project Leaders are responsible for the execution of a specific project and will be evaluated based on the result of the specific project and not how well it is preparing the way for later projects. It is important that the requirements related to the modularity are clearly specified in project description to ensure that the modular aspects will be covered and evaluated. This is not addressed specific in this project, but visualization how the modules in the first Frequency Variant will affect future variant can be valuable in the project definition.

3.1.4. Development

Development is executed by multiple sub-groups with different responsibilities, see Table 4. The modular approach will bring great advantages to the development when reuse can reduce the total required development time, but will also add more complex requirements related to compatibility for modules used in multiple variants.

Table 4. Sub-groups within the development

Sub-group	Responsibility
Digital	Digital side of the RU
TX – PA	Power Amplifier of signal for transmitting
TX - Up conversion	Up conversion from baseband to carrier frequency
RX - Front end	Filter and low-noise amplification
RX - Down conversion	Conversion of received signal from carrier frequency to base band
Filter	Filters used in both RX and TX
Antenna interface	Interface between RU and antenna
Mechanics	All mechanical design
DC/DC	Voltage conversion to multiple desired levels
Frequency generation	Clock reference

To ensure that the developed modules are prepared for future usage in other planned variants, it is important that everyone have a common understanding of where and how the modules will be used. Visualizations can facilitate this common understanding.

It is a challenge to argue for the extra effort a compatible module will required within development. By using clear visuals where the reduced resource usage and total development time due to modularity would be a powerful tool to argue and implement the organizational changes that are needed to fulfil the modular plan.

Ericsson is using shelf development to make the development of similar modules more effective. An example is that the development of power amplifiers (PA) for multiple frequencies benefit from being developed in parallel by the same team. The total development time for one team will be less than letting different teams do it, since problems solved for one PA can be applicable for the rest.

3.1.5. Integration

Integration is responsible for the integration of all hardware modules and software into a complete product. Integration would benefit from reusing part that previously have been integrated and are well-known. On the other hand, a modular approach might lead to more complex interfaces, due to the separation of common and shared features. This could therefor lead to bigger challenges within integration and also lead to greater cost.

The usage of shelf development might lead to a time-gap between the end of development and the start of integration of a variant. This might lead to confusions when problems occur in the integration part and no dedicated team is working with the development any longer. To reduce the risk of this, interfaces for modules need to be well specified, documented and verified. By shorten the time between the end of a shelf project and the integration of the modules, the risks will also be reduced.

3.1.6. Verification

Verification is responsible for verifying that developed products are fulfilling functional and performance requirement. The advantages and disadvantages of modularity described for integration are also applicable for verification.

3.1.7. Sourcing

Sourcing is responsible for buying all needed material and components to produce the product family and negotiate to get as high quality as possible for lowest price possible. A modular approach will very likely decrease the number of different components and increase the volume of each component, which gives lower sourcing costs. But it could also result in higher sourcing cost since a shared module might be more complex than multiple unique modules and require more and/or more complex components. Much more in-depth analysis needs to be done in this area to find how sourcing are affected in details. But a visual representation of how components are reused can be a good help in the discussion how to find the best modularity.

3.1.8. Supply

Supply is affected by modularity in multiple ways. One example is how modularity affects the goal to deliver ordered products as fast as possible, without having too much capital bounded in stock. If it is possible to reuse big and expensive modules the total bounded capital will decrease with the same flexibility. And if the unique modules are relatively cheap, they can be stocked and, for a very low costs, reduce the delivery time. Figure 26 shows a simplified production of 3 variants of a product, P_a, P_b and P_c. In this case, there is no modular approach where all variants have unique cards that will be assembled in the same enclosure. When an order arrives the production starts at T_0 with the production, test of the card and lastly assembled before it is tested and shipped to the customer at T_2. To shorten the lead time from order to delivery, supply could stock unique cards at T_1, but this would result in very high capital binding.

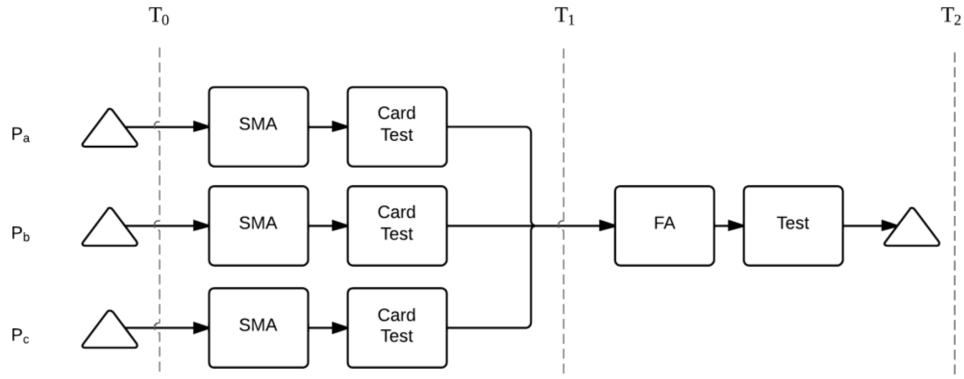


Figure 26. Supply chain without modular approach

With a modular approach the cards have been split into two separate modules, one common, M1, and one unique, M2, see Figure 27. The common module can now be stocked at T_1, since it will be used in all three variants of products and the flow will be much higher. If the unique modules are cheap, they can also be produced for stock at T1, without binding a lot of capital. By doing this, the delivery time can be reduced since the order only needs to escalate back to T_1.

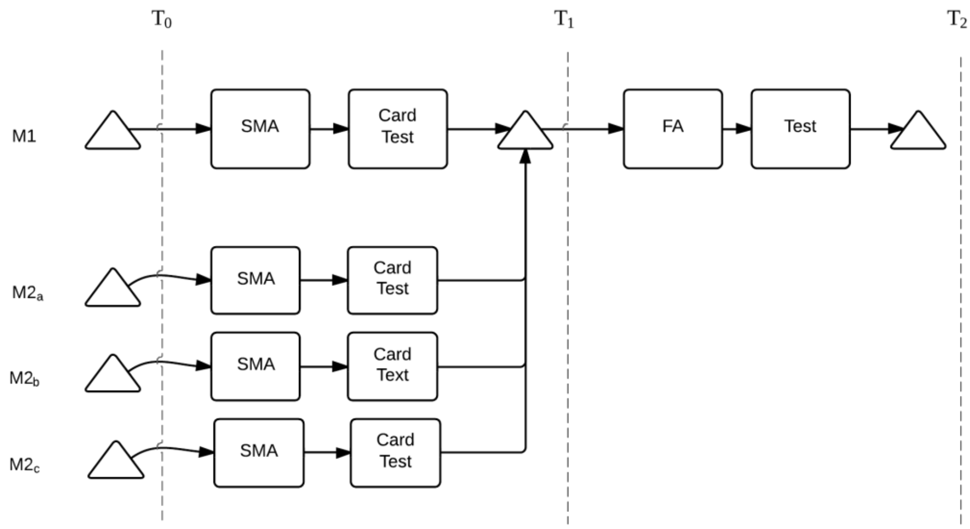


Figure 27. Supply chain with modular approach

Another aspect that can be affected by modularity is the number of unique components in the surface mounting. A production line has a maximum number of trays to store unique parts that are available for the robot to place. If the total number of unique components within Frequency Variants is within the maximum, the line can easily be changed to a new variant without manual adjustments of the trays. Since both these aspects are hard to estimate early in the planning phase more investigations are needed to see if it would benefit from visualization.

Supplying old products can result in increasing costs since redesign might be needed if components aren't available any more. A production line is often customized to produce variants within the same platform. If the number of produced variants within a platform decreases, the cost for having the line up and running will increase per produced unit. With modularity the extra development cost for replacing an old Frequency Variant will be less than developing it from scratch. This can motivate development of replacement variants within the new platform since the entire old platform can be canceled. A visualization of estimated development cost for a replacement variant would help the discussion if it is profitably to close an old platform.

Carry-over of modules is in general good from a supply perspective. The production and test is already in place. But since the technology change so fast components might not be available as long as desired and redesign is needed without any increased performance or saved cost. By visualizing the time-plan for both variants and containing modules it would be possible to see if some carry-over-modules will expire before the planned variant.

3.1.9. Maintenance

To reduce the manufacturing cost, iterations of improvement projects are executed where new technology is introduced. One example is when an early version contains multiple FPGA to complement one ASIC and a maintenance project enables all features to be implemented in one ASIC. This will eliminate a number of FPGAs and decrease the overall manufacturing cost. When using a modular approach with clear defined interfaces the replacement work of one module will be much easier than a fully unique and integrated product. If a module is reused amongst multiple variants the improvement can more easily be implemented in the other variants and improve the savings even further.

3.1.10. Repair

Since it is crucial that an operator's network is up and running, backup-units are available for the operators to replace broken RUs directly on site. The broken unit is shipped to one of Ericsson's repair centers where the unit is repaired, tested and calibrated. This principle is called "swap-and-repair". A modular approach does in general simplify the repair process of products. A modularized product family enables replacement of whole modules and if variants share modules, the stock can be reduced. This shortens the repair time and reduces the binding of capital.

3.2. Concept Design of Views and Decision Support

Based on interview with key stakeholders multiple aspects have been identified. The following section will describe how these aspects can be visualized with one or multiple views. All equations, except equation (13), are self-derived based on information from interviews.

Some views are more relevant to certain stakeholders, while others are relevant for everyone involved in the entire product life cycle. A general mapping between stakeholders, relevant views and models are displayed in Table 5.

Table 5. Mapping between viewpoints and views and used models.

MODELS	Tree-View	Tree-View Expandable Table Sunburst	Organizational chart	Doughnut Chart	Interface Matrix	Stacked Bar Chart Stacked Column Chart Resource Table Bubble Chart	GANTT Chart	Stacked Area Chart
VIEWS	**Product Family Structure**	**Contains-of**	**Part-of**	**Reuse**	**Interfaces**	**Resource Usage**	**Time plan**	**Work load**
VIEWPOINTS								
Product Management	✓	✓		✓		✓	✓	✓
Product Planning	✓	✓	✓	✓		✓	✓	✓
Project Leader	✓	✓	✓	✓	✓	✓	✓	✓
Development	✓	✓	✓	✓	✓	✓	✓	✓
Integration	✓	✓	✓	✓	✓	✓	✓	✓
Verification	✓	✓	✓	✓	✓	✓	✓	✓
Sourcing	✓	✓	✓	✓			✓	
Supply	✓	✓	✓	✓	✓	✓	✓	
Maintenance	✓	✓	✓	✓	✓		✓	
Repair	✓	✓	✓	✓	✓		✓	

3.2.1. Product Family Structure

In the near future Ericsson will have approximately 500 variants of RUs. Before trying to communicate the detailed information and aspects of modularity, it is important to get a common understanding of the overall structure. What different product groups exist and what different variants will be developed within each product group? By visualizing the structure in an interactive organizational chart the user can both se the overall structure and at the same time zoom in on a specific product group for more detailed information, see Figure 28.

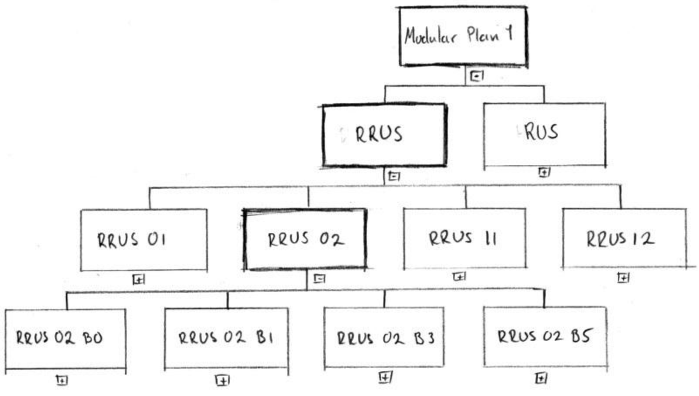


Figure 28. Sketch of an interactive organizational chart displaying product structure.

The structure contains of multiple levels of items, where *Modular Plan* is on the very top level, followed by *Product Type, Product* and *Frequency Variant*. Each Frequency Variant contains of elements of different types; *Subassembly, Physical Element, Logical Element* and *Component*.

3.2.2. Contains-of relations

When having a common understanding of the whole product family, it is also important that everyone is able to understand what elements are used to build up different Frequency Variants. Information about reused and new elements is also important. This contain-structure can be visualized in an interactive organizational chart, see Figure 29.

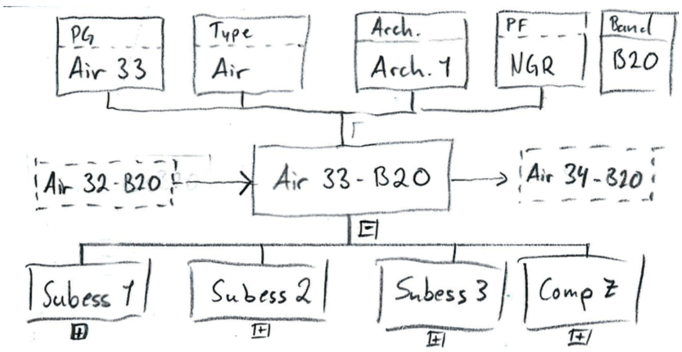


Figure 29. Sketch of interactive organizational chart for contain-structure.

If more detailed information is desired an expandable contain-table can be used to view multiple attributes on containing elements, Figure 30

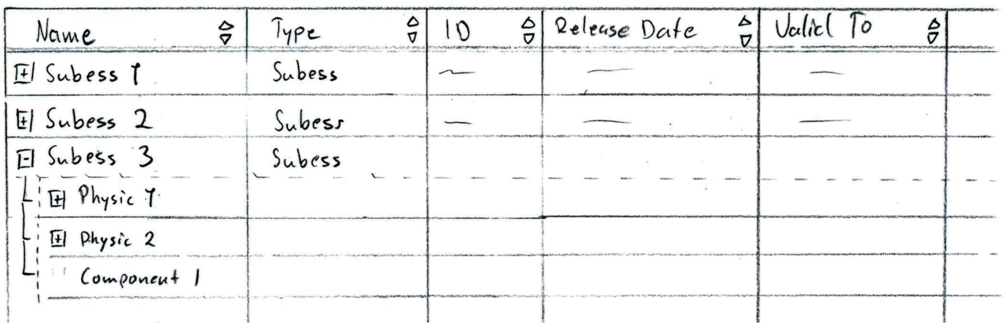


Figure 30. Sketch of expandable contain-table with more detailed information.

An alternative way to visualize how different variants are built up by elements is by using a sunburst, see Figure 31, where each sector is representing a variant with the elements building up its basic features. Each unique element has its own colour and by hover an element all other entities of the same element are highlighted. Shared elements amongst variants are visualized in a good way and give a good overview and the ability to make a first comparison between different modular plans.

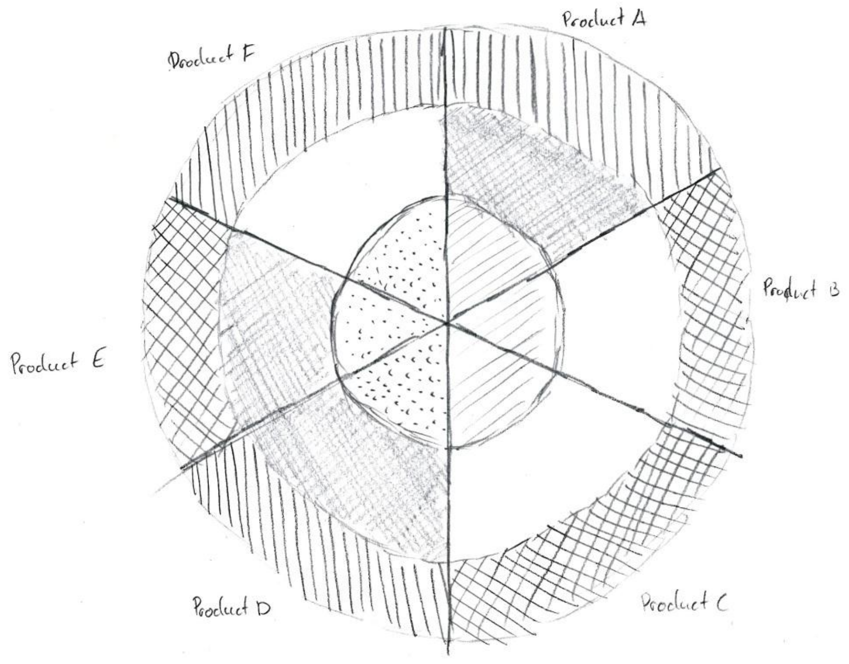


Figure 31. Sketch of a segmented sunburst visualizing the containing elements for different variants.

3.2.3. Part-of relations

It is not only important to visualize modularity from a Frequency Variant perspective, but also from an element perspective. Starting from an element, it is desirable to see where and how it is used in multiple Frequency Variants. This is a good start to examine what requirements apply to the element. This can also be visualized in an interactive organizational chart, see Figure 32

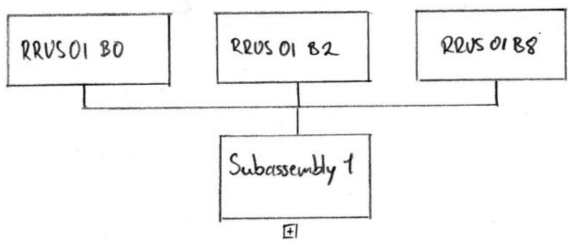


Figure 32. Sketch of interactive organizational chart for part-of-structure.

3.2.4. Reuse

Reuse is at Ericsson defined as *shared modules within a platform* and Carry Over is defined as *shared modules between different platforms*. Both reuse and carry over will further on be considered as reuse.

The reuse of elements will have different impacts depending on the level in the product structure. When reusing a Subassembly or a Physical Element the benefits can be seen within development, I&V, supply and sourcing. The reuse of individual Components within different designed Logical Element mostly will benefit the sourcing and supply. Therefor three different reuse indicators have been identified to indicate different aspects of the reuse; Subassembly reuse, Physical reuse and Logical reuse.

Two different KPI have been derived for each level, *numerical reuse* that considers the number of elements that are reused from previous variants and *development time reuse* that considers the development time saved by reuse.

Numerical reuse for sub-assemblies, $R_{sub,num}$, is between zero (0) and one (1) and is defined as

$$R_{sub,num} = \frac{\sum_{k=1}^{n_{sub}} r_k}{n_{sub}} = \frac{\text{number of reused sub-assemblies}}{\text{total number of sub-assemblies}}, \quad (7)$$

where n_{sub} is the number of sub-assemblies and r_k is one (1) if sub-assembly *k* is reused and zero (0) if it is new. Development time reuse for subassemblies, $R_{sub,time}$, is also between zero (0) to one (1) and is defined as

$$R_{sub,time} = \frac{\sum_{k=1}^{n_{sub}} t_k \cdot r_k}{\sum_{k=1}^{n_{sub}} t_k} = \frac{\text{development time for reused sub-assemblies}}{\text{total development time for sub-assemblies}}, \quad (8)$$

where t_k is the development time for sub-assembly *k*.

Physical reuse ($R_{phy,num}$, $R_{phy,time}$) and Logical reuse ($R_{log,num}$, $R_{log,time}$) are calculated in the same way with the only difference that the number of sub-assemblies, n_{sub}, is replaced with the number of physical elements, n_{phy}, and the number of logical elements n_{log}.

Reuse can be visualized with doughnut charts where the percentage of reuse time is indicated, see Figure 33. The reused number of man-weeks is also displayed together with the total amount of man-weeks. In the same way the number of reused elements is displayed with the total amount of elements.

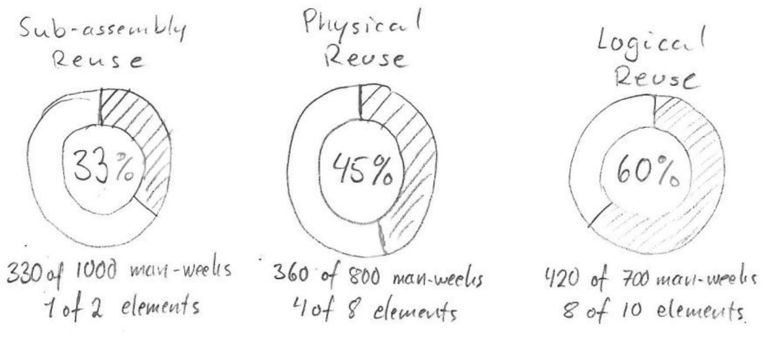


Figure 33. Sketch of doughnut chart representing reuse.

3.2.5. Interfaces

From reuse perspective, all common functions would preferable be put into shared elements. But that approach might lead to a lot of complex external interfaces between elements. And since it is easier to handle complex interfaces within an element (internal interface), it is important to handle these conflicting interests in a good way. A visualization of interfaces are therefore desired.

Interfaces are categorized as one of the following:

- Digital signal
- RF
- Voltage
- Thermic
- Mechanical

In the same way as development times are estimates, the complexity of an interface is estimated on a scale:

1. Minimal
3. Easy
5. Medium
7. Challenging
9. Very Complex

The interfaces are visualized in an expandable matrix where both external and internal interfaces are displayed, see Figure 34.

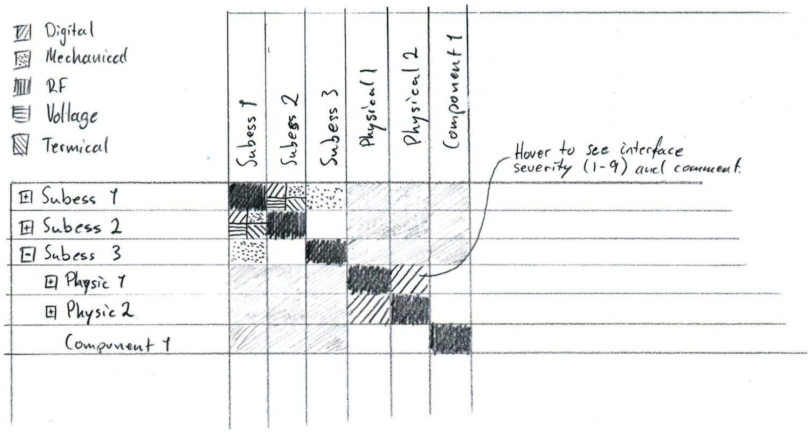


Figure 34. Sketch of matrix visualizing external and internal interfaces.

3.2.6. Resource Usage and Development Time

How modularity affects the resource usage is, based on interviews with stakeholders, one of the most important aspects. The major question is what modularity plan that requires least resources for the whole product line. It is important to visualize estimated resource usage of each Frequency Variant and also see how estimations from lower elements together give the total outcome. To do this multiple models have been developed.

The considered resources are displayed in Table 6. All resources are related to development, e.g. the implication of *Supply* is resources needed to make a Frequency Variant ready for production and not related to the resources for producing a Frequency Variant.

Table 6. Resources used in tool

Resource
Filter
Digital Asic
PA
Low Power RF
Mechanics
Supply
Integration
Verification

The required usage of a resource for developing an element, $T_{r,i}$, is derived as the summation of the resource usage of all containing elements, plus resource usage specific for the element itself, $t_{r,i}$, where r is the resource and i is the element. The definition is

$$T_{r,i} = \sum_{k=1}^{n} \left(T_{r,k} \cdot \left(1 - r_k \right) \right) + t_{r,i} , \qquad (9)$$

where n is the number of all containing elements and $T_{r,k}$ is the resource usage for each containing element. r_k is one (1) if the element is reused and zero (0) if it needs to be developed.

The total development time for an element, T_i, is derived as the summation of all resource usage defined as

$$T_i = \sum^{r} T_{r,i} . \qquad (10)$$

By using modularity the resource usage for developing an element can be spread amongst multiple variants and reduce resource usage per variant. The total development time per Frequency Variant, $T_{variant,i}$, is defined as

$$T_{variant,i} = \frac{T_{r,i}}{n_{variant,i}} , \qquad (11)$$

where $n_{variant,i}$ is the number of variants using element i.

Resource usage and total development time can be visualized in multiple ways. The most basic model is by using a stacked bar when displaying it for a specific element, see Figure 35.

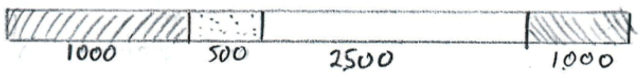


Figure 35. Sketch of resource usage and total development time for an element

To understand the resource usage it is not only important to visualize the absolute estimated value, but also show where the values come from. Especially when reasoning about the accuracy of an estimation. To do this an expandable resource table can be used to show how the calculation is done. Figure 36 shows an example of how the total resource usage of an element is the summation of containing elements resource usage plus the unique resource usage for the element itself.

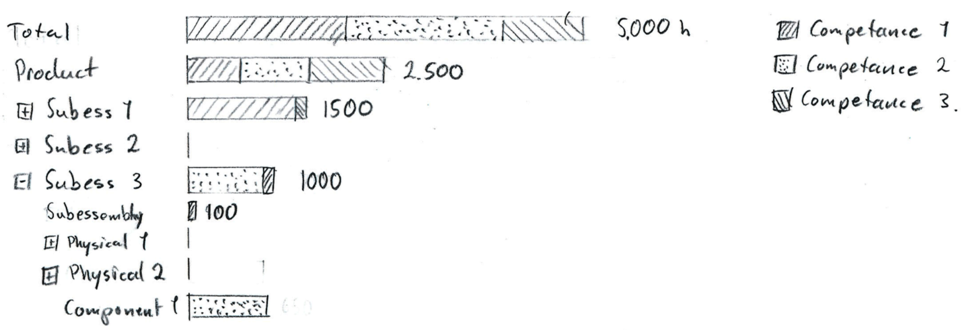


Figure 36. Sketch of an expandable table to visualize the calculation behind the total resource usage.

When looking from a product perspective it is interesting to visualize the resource usage per variant. This can be visualized by using a stacked column chart, see Figure 37.

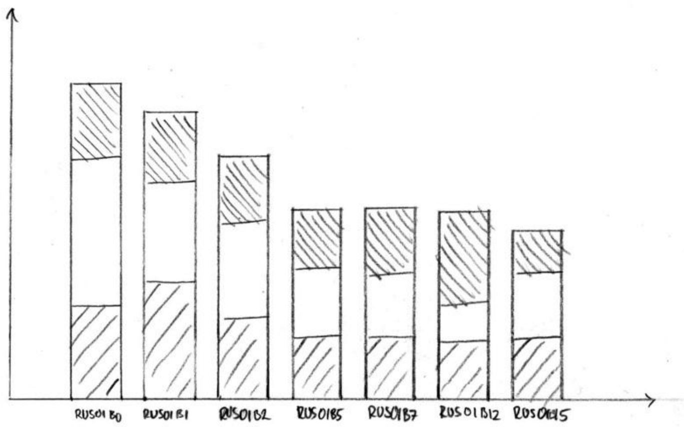


Figure 37. Sketch of stacked column chart.

One level higher, on a product type level, bubble chart can be used to visualize how the total time differs between multiple products and its variants, Figure 38

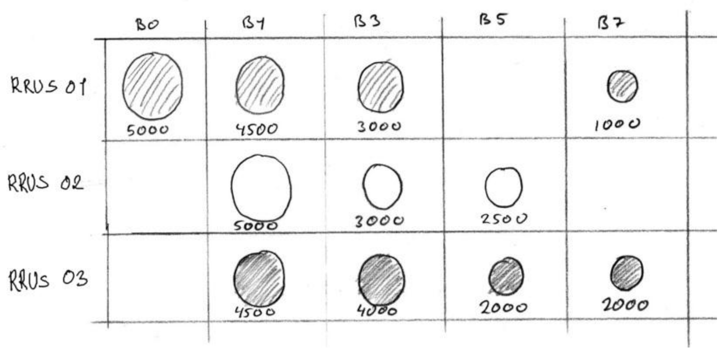


Figure 38. Sketch of bubble chart to visualize the total time for multiple products and its variants.

3.2.7. Time plan

The development time for the first variants, with intention to reuse elements for later variants, will be longer than a unique developed variant. This will obviously change the release plan where the time-to-market for the first variants will be extended, but shortened for later variants.

A visual representation of the time plan, including development-start-date, release-date and valid-to-date for the product family would help the product management and product planning in their decision on selecting modular plan. GANTT-charts, see Figure 39, are commonly used to visualize time-plans and suites this purpose well. By using an expandable GANTT-chart, the containing elements time-plan can also be visible if desired.

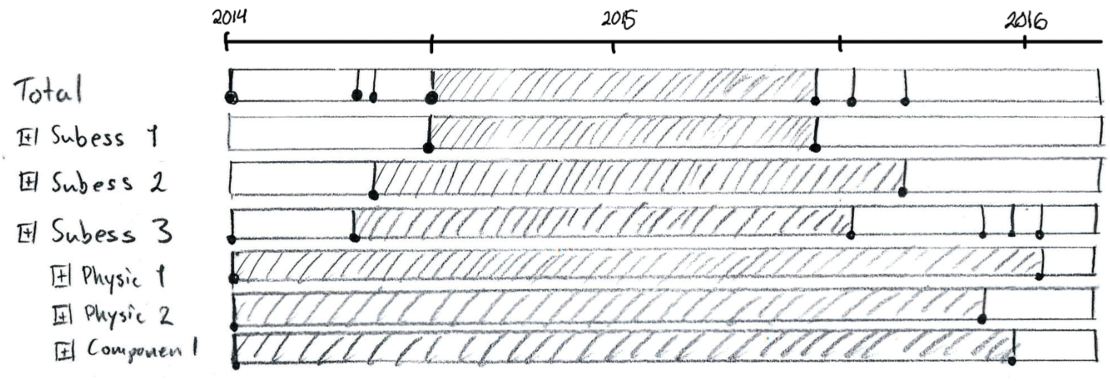


Figure 39. Expandable GANTT-chart.

There are two other important aspects to high-lights in the time-plan. The first one is to indicate if a containing element has a shorter valid-to-date than the element and/or Frequency Variant itself. This might occur when reusing an element from previous platform with the consequence that the module might need a redesign due to obsolete components.

The other aspect is the time-gap between development and integration of an element, with the risk described in 3.1.4. By making it visual for everyone this can be considered and reduce the risks.

3.2.8. Work load

Another view of the resource usage is how the workload for resources is distributed over time. Ericsson has other tools for detailed time-planning, but it is desired to visualize how the workload will vary over time and not least indicate if there are time-slots where a modular plan will extend the available resources at a certain time.

The workload is calculated based on the estimated resource usage and time-plan and can be visualized at different levels. Two examples are how the workload for filter design will change for a whole modular plan and how all resources' workload varies for a specific Frequency Variant.

The workload for a resource coming from only one leaf element (no containing elements), $l_{r,i}(t)$, at time t is derived as

$$l_{r,i}(t) = \frac{t_{r,i}}{w_{r,i}} \chi_A(t), \tag{12}$$

where $t_{r,i}$ is the resource usage, $w_{r,i}$ is the number of weeks and $\chi_A(t)$ is a rectangular function defined as

$$\chi_A(t) = \begin{Bmatrix} 0 & \text{if } t < t_{start,r,i} \\ 1 & \text{if } t_{start,r,i} <= t <= t_{end,r,i} \\ 0 & \text{if } t_{end,r,i} < t \end{Bmatrix}. \tag{13}$$

$t_{start,r,i}$ is the start-time and $t_{end,r,i}$ is the end-time of the resource usage. A plot of $\chi_A(t)$ is shown in Figure 40.

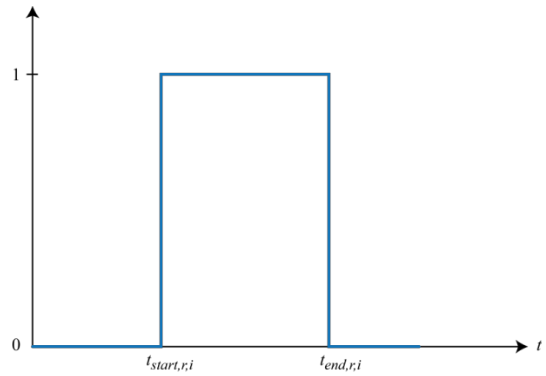


Figure 40. Plot of the rectangular function $\chi_A(t)$

For elements containing sub elements the total workload for resource *r*, at time t is defined as

$$L_{r,i}(t) = \sum_{k=1}^{n}\left(L_{r,k}(t)\cdot(1-r_k)\right) + l_{r,i}(t), \tag{14}$$

with the same reasoning as for equation (9).

The total workload at time t for element *i*, $L_i(t)$, is the summation of all resource workload defined as

$$L_i(t) = \sum^{r} L_{r,i}(t). \tag{15}$$

Workload can be visualized in a stacked area chart with time on horizontal axis and required number of workers on vertical axis, see Figure 41. Each area is representing the workload of a specific resource.

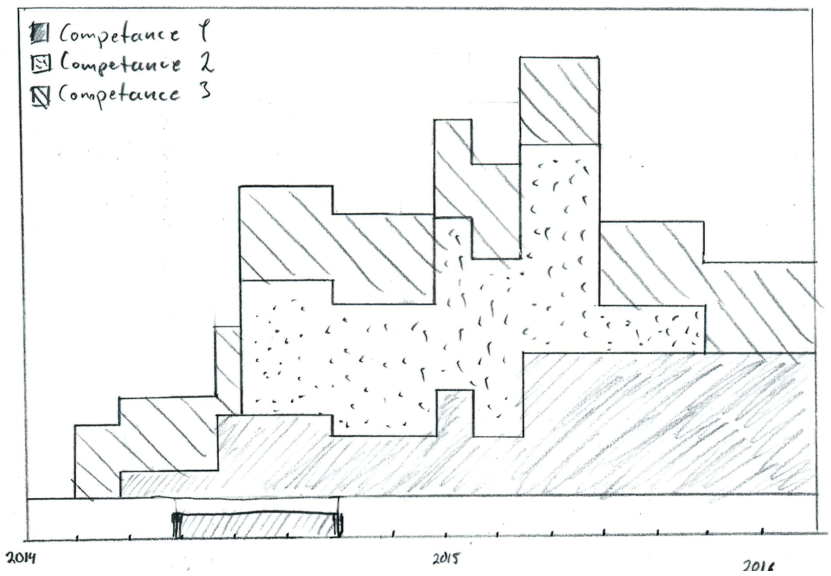


Figure 41. Sketch of workload chart for an element.

An alternative model is to let areas in the chart represent containing element or variants of a product. By doing this it is easy to see the reasons behind the workload and how it will vary in time.

3.3. *Elicitation of requirements for the visualization tool*

Except requirements related to different stakeholders, requirements were also identified related to how the visualizations are intended to be used. All requirements are presented in Appendix 1, but this section contains a highlight of the most important requirements and an analysis of available tools and frameworks to implement a visualization tool.

3.3.1. Highlights of most important requirements

The tool is intended to be used in two phases; (1) Selection of the best modular plan within the pre-development and (2) communicating selected modular plan within the whole organization. In both phases the number of editor-users who will change data will not be more than three persons. The number of read-users will be approximately 25 in selection phase and potential up to 300 in later phases. Therefor there are different requirements for editing information and access views in the tool.

There is also a desire to be able to customize the views in real time to highlight specific aspects. An example is the ability to select displayed resources in the workload-chart and explicit specify the time-frame.

The views shall be accessible for multiple users within Ericsson's internal network and not require any installations not provided for free from Ericsson IT-service. Despite that different views are visualizing different aspects of the modularity, it is important that views are generated from the same source of data. Therefor the tool needs to support a centralized storage of data where all views are generated from.

The first version of the tool focuses on how modularity affects development time. If the prototype works well, there is a desire to be able to develop the tool with features covering more aspects. Therefor the tool is desired to be built on well-known technologies that future developers can easily extend.

Data structure

A suggested data structure was developed that supports generation of data used by all views. Figure 42 shows a simplified version of the structure. This was further developed together with Holm (2014), responsible for developing the back-end. The complete data structure can be found in Appendix 2.

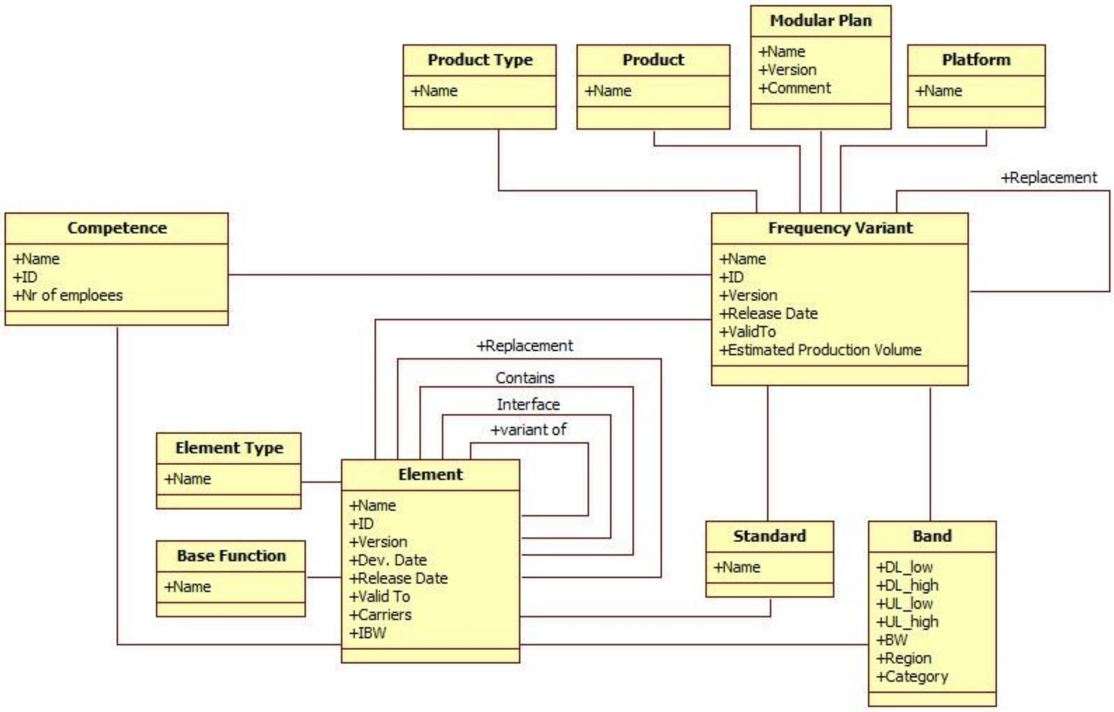


Figure 42. Simplified data structure to support data for all desired views

Frequency Variant and *Element* are the two most central objects in the structure. Except by the attributes, a Frequency Variant is categories by relations to different objects, e.g. *Band*, *Product* and *Modular Plan*. *Elements* can be connected to multiple instances of *Frequency Variants*. This enables one *Element* to be used in multiple *Frequency Variants* without duplicated data.

Sub-assemblies, physical elements, logical elements and component are all *Elements*, but categories by a relation to *Element Type*. Both *Frequency Variant* and *Element* are connected to supported *Band* and *Standard*, e.g. GSM, WCDMA and LTE. Multiple instances of *Competences* can be connected to both *Frequency Variants* and *Elements*, where the estimated resource usage, start-date and end-date are specified on the relation. By doing this, it is possible to both see all *Competences* connected to an *Element* and all *Elements* requiring a specific *Competence*.

3.3.2. Selection of framework for visualization tool

With the requirements as base, an investigation was done amongst existing tools to find a good candidate. Most visualization tools are targeting business intelligence and meet a lot of the requirements related visualizing resource usage and development time. But the lack to visualize product structure and interfaces will exclude them as an option. MetaEdit+ contains good ability for product structure visualization, but will require integration with other tool to visualize resource usage and time-planning. Due to the lack of available off-the-shelf tool the option of developing an own tool was selected.

Web technologies have been selected for developing the tool, since it fulfills all the requirements and have some big advantages compared to other options. The ability to let multiple users access a web application without any extra installation by using a regular web browser is strongly favoring web technologies compared to desktop applications. There are also already available JavaScript visualization libraries that, with minor customization, meet the requirements of all views and models. A solution based on Microsoft Office would require a lot more development to generate desired visualizations. The positive trend for web-application is also a good argument to enable future development.

3.4. Prototyping

As a proof of concept, a front-end web-tool has been developed. This section describes the overall technical implementation together with all implemented views and models.

3.4.1. Technical Overview

The tool is built by customizing BucketAdmin ("Bucket Admin," 2014), a web template based on Bootstrap, together with multiple other JavaScript libraries for realizing the different views.

AngularJS ("AngularJS," 2014) is used as a framework to support interactivity and connect the front-end with the back-end and still be able to develop them separately. This is possible since data is sent between front-end and back-end by using Json-structures, see Figure 43. The back-end of the system was developed by Holm (2014) and will not be described in detail. The tool was developed and verified by using static files containing example data. The tool can though easily be integrated with back-end since the interfaces are well defined.

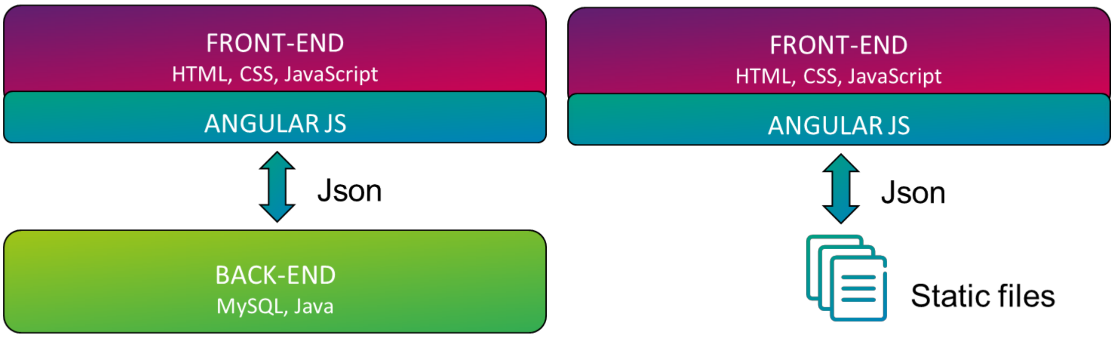


Figure 43. Framework for integrating front-end and back-end

The tool has three major sections, top-bar, navigation-area and content-area, see Figure 44. Top-bar contains logo and user information, navigation-area the main navigation and content-area the selected content.

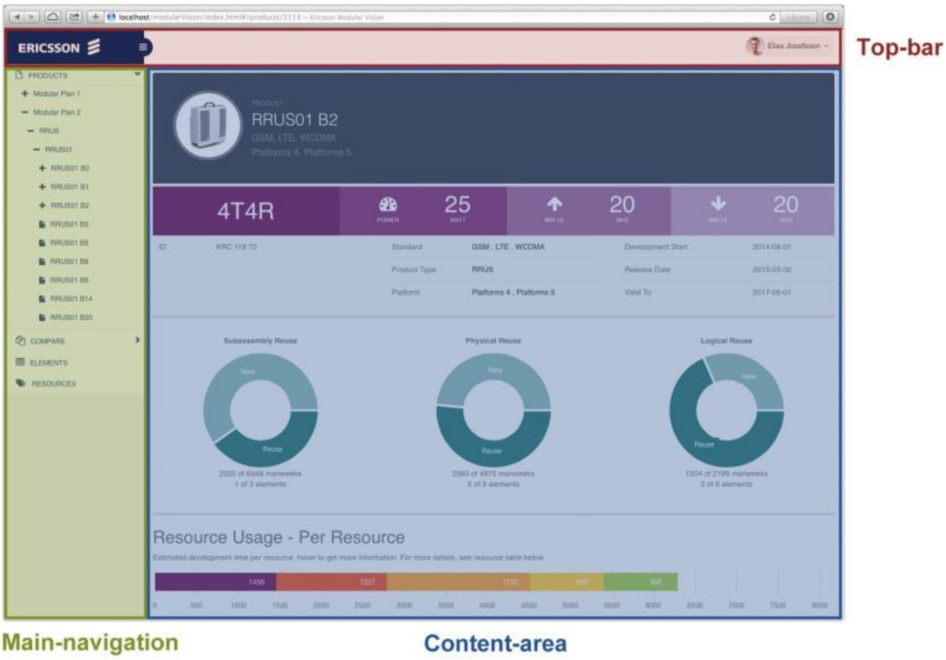


Figure 44. Overall structure of tool.

Main navigation contains four main menus that give different entry points to access desired information. *Products* contain all available modular plans and are displayed in an interactive tree view. The tree view contains all levels from Modular Plan to Logical Element, see Figure 45.

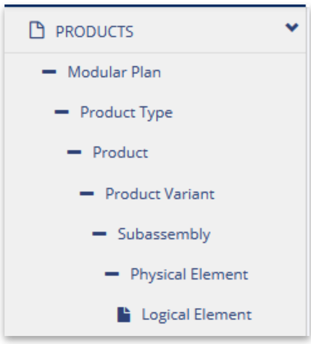


Figure 45. Tree structure for products.

Each menu opens up the views for the selected item in the content-area, see Figure 46.

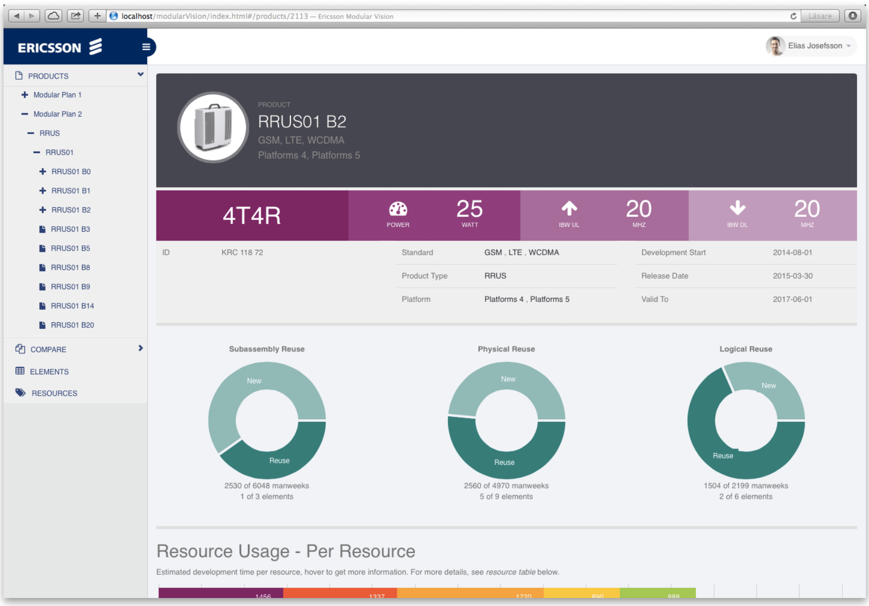


Figure 46. An example of Frequency Variant page with its views.

Compare contains the same modular plans with corresponding tree view as the product menu, now with a checkbox per item for compare selection, see Figure 47.

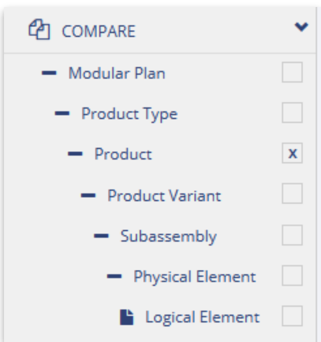


Figure 47. Tree view for compare.

Selected items are grouped per modular plan and comparison views appear in the content-area, see Figure 48.

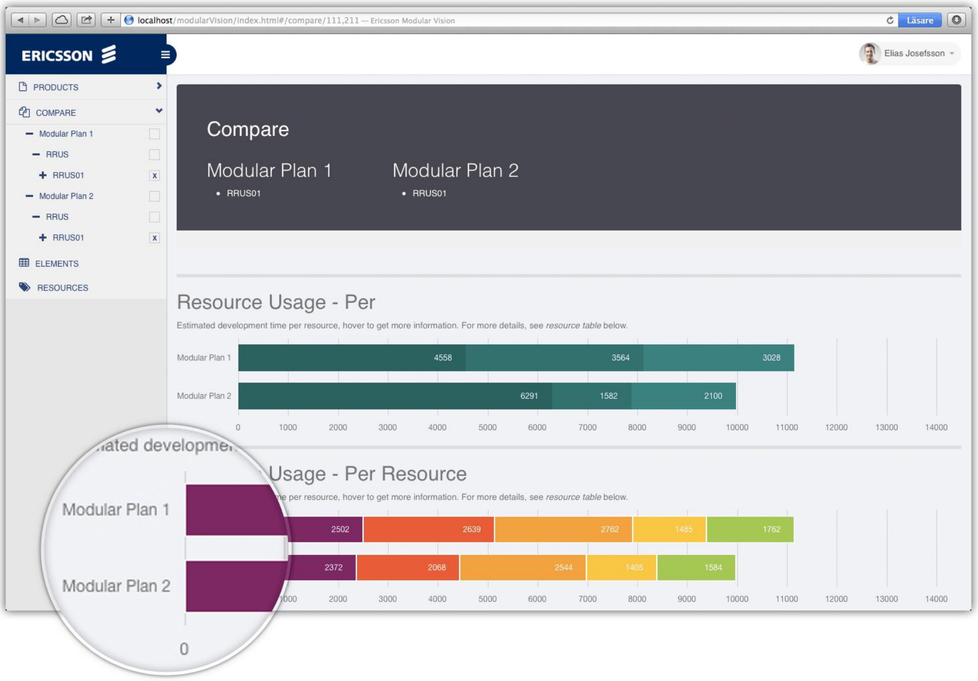


Figure 48. An example of a comparison of a product between two different modular plans.

Elements leads directly to the element page containing a searchable and sortable table with all available elements in the tool, see Figure 49.

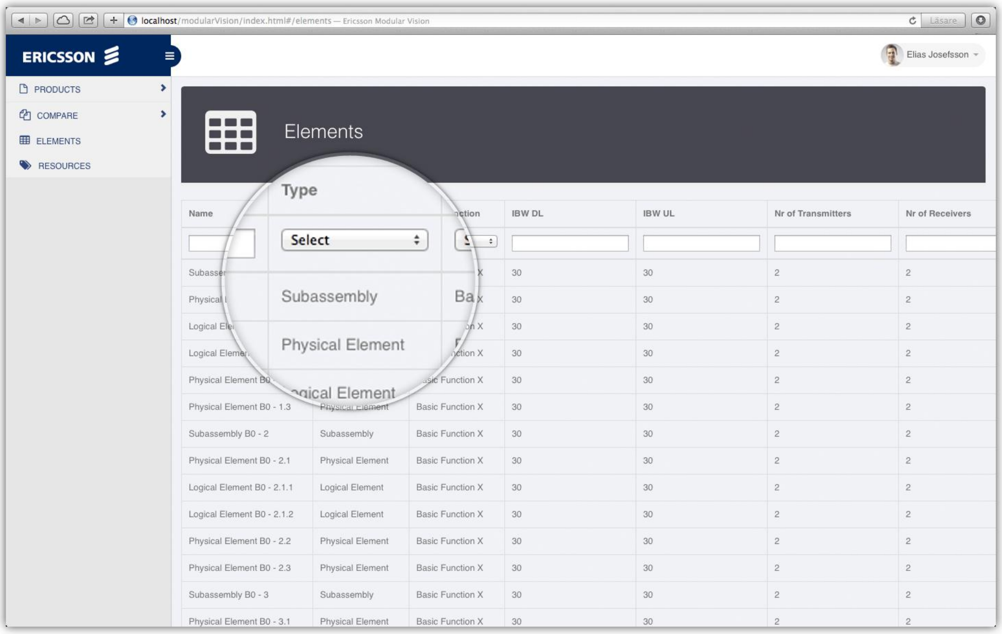


Figure 49. Searchable and sortable element table.

Resources contain menus to all available resources linking to views with each resource as base. These views are not yet implemented in the tool.

Multiple models are implemented to give as good visualization of the modular information as possible. A brief description of each model and how it is implemented are presented. All models are not relevant for all levels in the structure. A mapping of models and levels are presented in Table 7.

Table 7. Mapping of models and levels.

MODELS \ LEVELS	Modular Plan	Product Type	Product	Frequency Variant	Subassembly	Physical Element	Logical Element
General Information	✓	✓	✓	✓	✓	✓	✓
Reuse				✓	✓	✓	
Development Time – Stacked Bar-chart	✓	✓	✓				
Development Time – Bubble-chart		✓					
Resource Usage – Stacked Bar-chart	✓	✓	✓	✓	✓	✓	✓
Resource Usage – Stacked Column-chart	✓	✓	✓				
Product Family Structure	✓	✓	✓	✓	✓	✓	✓
GANTT-chart	✓	✓	✓	✓	✓	✓	✓
Contain Table	✓	✓	✓	✓	✓	✓	
Resource Table	✓	✓	✓	✓	✓	✓	
Work Load – Per resource	✓	✓	✓	✓	✓	✓	✓
Work Load – Per Item	✓	✓	✓				
Interfaces				✓	✓	✓	
Sunburst			✓				

3.4.2. Information

General information is presented at the top of the page where most important information is highlighted in coloured boxes. The type of information adjusts to selected level. Figure 50 shows an example of the information displayed for a Frequency Variant.

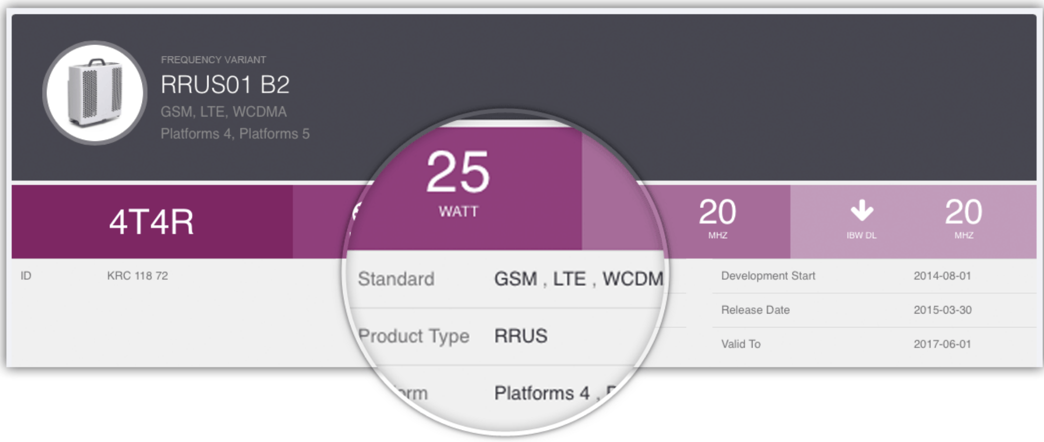


Figure 50. An example of general information for a Frequency Variant.

3.4.3. Reuse

The three reuse aspects; Subassembly Reuse, Physical Reuse and Logical Reuse are visualized in three different doughnut charts, see Figure 51. These are implemented by using the JavaScript library *ChartJS*.

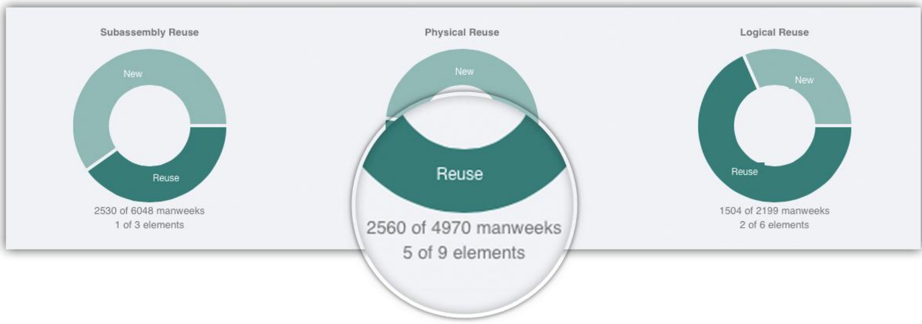


Figure 51. An example of reuse visualized for a Frequency Variant.

3.4.4. Development Time – Stacked Bar-chart

The total development time for an item, e.g. product, is a summation of the development time for all variants. This is visualized in a stacked bar-chart. Each variant's development time is represented by a coloured part of the bar-chart, see Figure 52. The bar-chart is implemented by using the JavaScript library *ChartJS*.

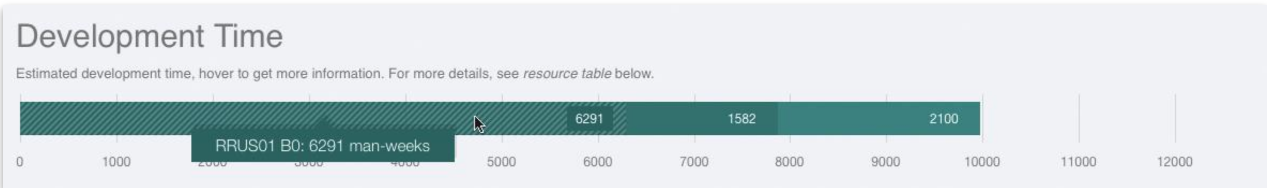


Figure 52. Development Time visualized in a stacked bar-chart.

3.4.5. Development Time – Bubble-chart

When looking at a Product Type-level a bubble chart is used to visualize how the development time differs between variants in different products, see Figure 53. This model is implemented by an own developed directory for Angular JS.

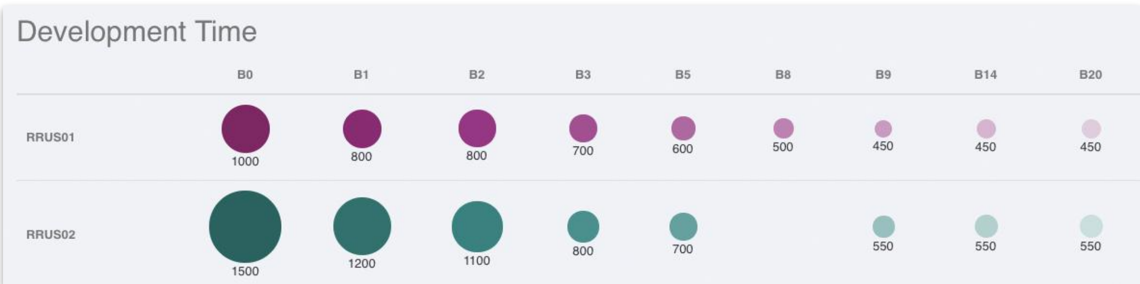


Figure 53. Development time visualized for variants of multiple products.

3.4.6. Resource Usage – Stacked Bar-chart

The resource usage for an item is visualized in different ways depending on the level. Used for all levels is the stacked bar-chart where each colour represents a specific resource, see Figure 54.

Figure 54. Resource usage visualized in a stacked bar-chart.

3.4.7. Resource Usage – Stacked Column-chart

How the resource usage differs between multiple variants or products are visualized in a stacked column-chart, see Figure 55. These column-charts are implemented with the library *ChartJS*.

Figure 55. Resource usage for multiple variants visualized in a stacked column-chart.

3.4.8. Product Family Structure

The product family structure is implemented with an interactive organizational chart that enables great overview of the structure and at the same time depth information. Figure 56 contains an example of the organizational chart on Frequency Variant level. RRUS 01 B2 is the Frequency Variant and all above levels are displayed together with platform and band. Previous version and replacement version is displayed to the left and right of the variant. The whole structure of elements contained in the variant is displayed beneath the variant. By clicking on a block the tree will rearrange to focus on the selected block. If interested in more detailed information, each block contains a link to the corresponding information page.

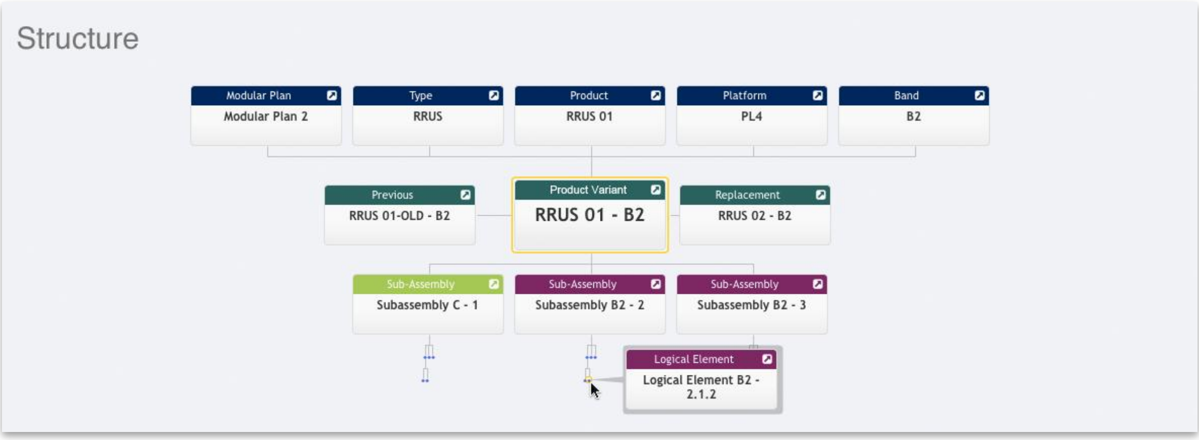


Figure 56. Organizational chart for a Frequency Variant.

The colour-coding, used to increase the readability, are presented in Table 8.

Table 8. Colour-coding used in organizational chart.

Blue	Modular Plan, Product Type, Product, Platform and Band
Blue-Green	Frequency Variant
Green	Reused Element
Purple	New Developed Element

3.4.9. Time Plan - GANTT-chart

The time plan is presented in an expandable GANTT-chart where each item contains two intervals: development and available, see Figure 57. By hover each interval, detailed information is displayed. The GANTT-chart is implemented by using the Angular directory *Angular-Gantt* (Schweighauser, 2014).

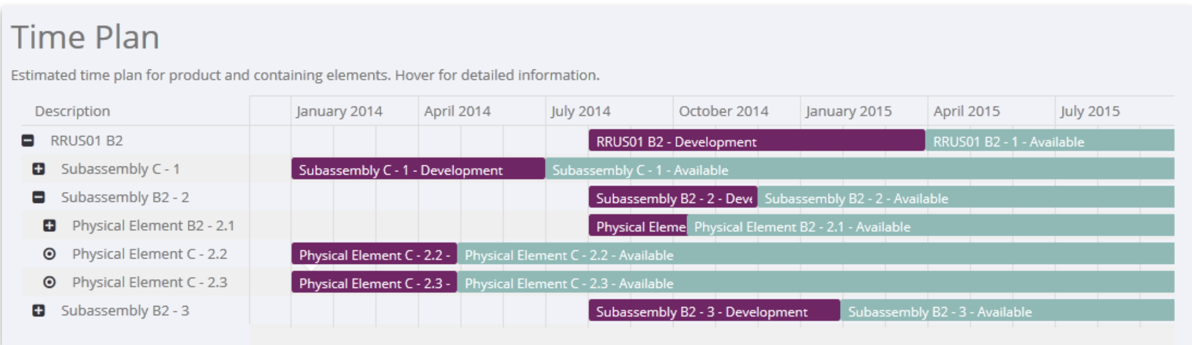


Figure 57. GANTT-chart visualizing time-plan for multiple elements.

3.4.10. Contain Table

By using an expandable contain-table, detailed information is available for all containing elements from multiple levels, see Figure 58. The implementation is a combination of an own developed directory for AngularJS and the library *jQuery treetable* (Boom, 2014).

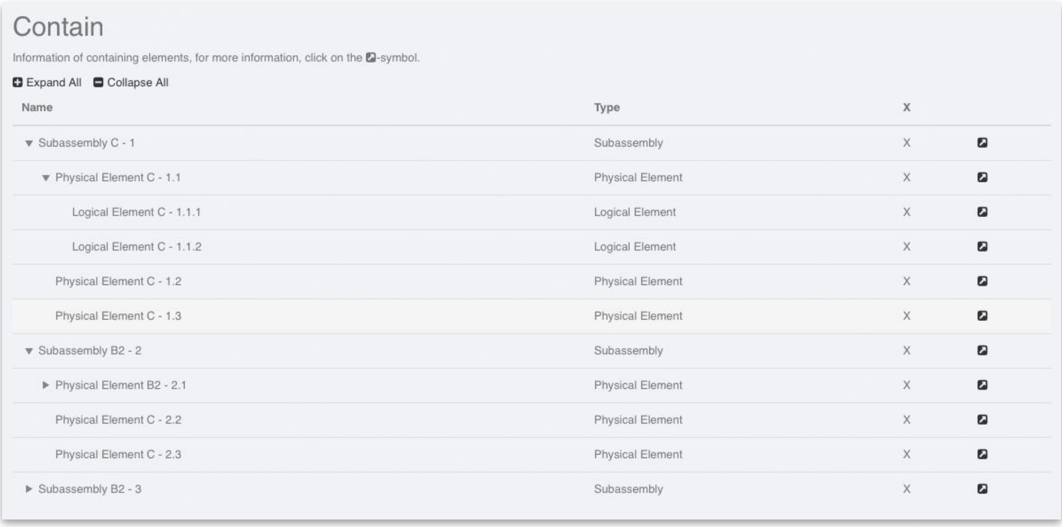


Figure 58. An example of a contain-table.

3.4.11. Resource Table

The resource table works in the same way as the contain-table, with stacked bar charts, representing resource usage, instead of multiple attributes per element. The total resource usage can be evaluated by looking at containing elements together with the unique development for the element, see Figure 59. This implementation is also a combination of an own developed directory for AngularJS and the library *jQuery treetable*.

Figure 59. An example of a resource table visualizing resource usage for multiple levels.

3.4.12. Work Load – Per Resource

The work load per resource is visualized in a stacked area-chart implemented with the library *ChartJS*, see Figure 60. The time-scale can be changed in real-time by changing the interval in the time-indicator at the bottom of the chart. It's also possible for the user to select what resources to display.

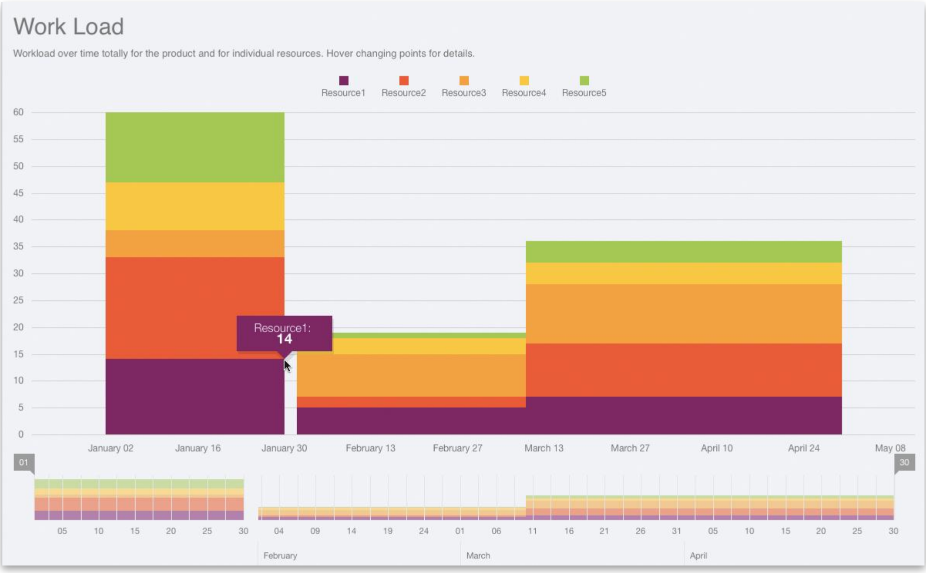


Figure 60. Work load where each stack is representing a resource.

3.4.13. Work Load – Per Item

This visualization works in the same way as previous work load, with the difference that each area is representing an item, e.g. containing Physical Elements is in a Subassembly, instead of a resource, see Figure 61. As in previous chart, it is possible to select what items to display for user customization.

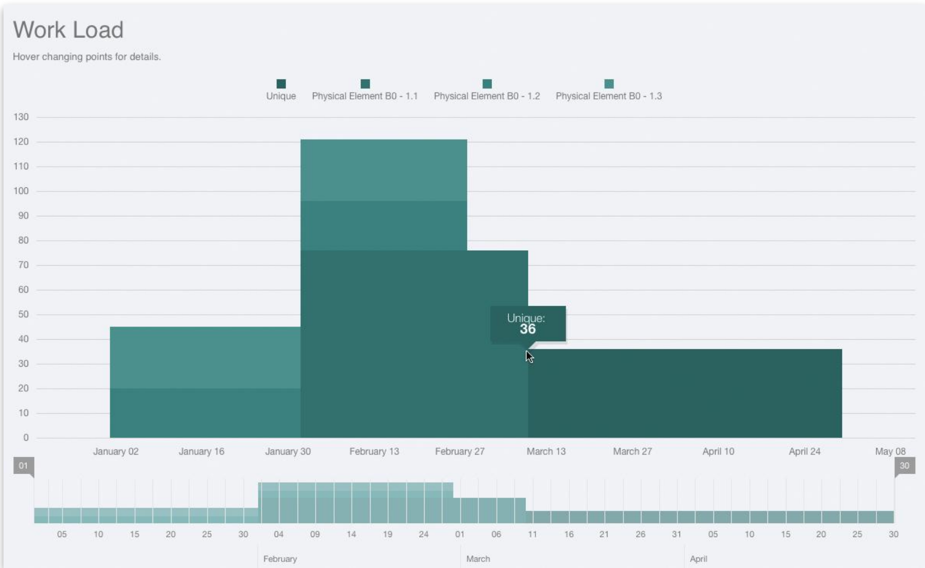


Figure 61. Work load where each stack is representing an element or item.

3.4.14. Interfaces

Interfaces within a variant or an element is visualized in a matrix containing colour-coded boxes to indicate the type and severity, see Figure 62. A tool-tip appears with detailed information when hover each box. The interface visualization is implemented with an own developed directory for AngularJS.

Figure 62. Interfaces represented with color-coded boxes in a dependency matrix.

3.4.15. Sunburst

The sunburst shows how elements are shared amongst multiple variants. Figure 63 shows an example of three variants of the product RRUS 01 and how some elements are shared. Each unique element has its own colour and by hover an element all other entities of the same element are highlighted. This is implemented by customizing a sunburst component (Rodden, 2014) based on D3.js.

Figure 63. How elements are shared amongst Frequency Variants is visualized in a sunburst.

3.5. Validation and Verification

The intention was to validate the tool with real historical data from a previous platform. Data regarding product structure is available down to a very low level, much deeper than intended in the tool. But since there are no available time-estimations for levels lower than sub-assembly, validation of the prototype is done with fictitious example data.

The example contains two modular plans, Modular Plan 1 and Modular Plan 2. Both contain the same products and Frequency Variants, where the variants in Modular Plan 1 contain only unique elements while Modular Plan 2 contains some shared elements.

The example was presented and discussed with selected stakeholders to validate how well the tool helped them to understand the modularity from their viewpoint. The overall feedback from stakeholders was that the tool successfully selects and visualizes relevant information. The interactivity gives users the ability to even further customize the views to make even more focused analysis. The different visualizations are simple and not really something new and similar static views have been used previously within Ericsson. But the combination of a big shared set of data with customized simple interactive views is the real success with this tool.

Since the example data is fictive, it is not validated how well the tool would work in a real development process. As with many other tools and models, what is coming in is also coming out. This means that if there are big uncertainties in the estimations inputted, the visualized result will also be uncertain.

All the functional requirements specified in Appendix 1 have been verified with satisfactory results. Since the implementation of the back-end was not ready during validation, all technical requirements have not been verified. The tool is available within Ericsson with the ability to make centralized updates, but since it only contains static example data, the requirement T3 is not yet fulfilled.

T3) The views shall be generated from a centralized set of data.

3.6. Derived Requirements for Back-End

The back-end of the prototype is developed by Holm (2014). Except the general requirements, the Json-structure defining the interface between front and back-end has been used as the input for the back-end. The Json-structure can be found in Appendix 3. More detailed information about the back-end can be found in the report for Holm's thesis (Holm, 2014).

4. DISCUSSION AND CONCLUSIONS

4.1. Discussion

This section contains a discussion about the process and the result of the project.

4.1.1. Process

The challenge to find the optimal modularity is a very huge and important task. Multiple researchers have focused on different aspects of the problem and give multiple suggestions for solutions. Ericsson has also previously addressed the problem from different angles. A lot of time was spent in the beginning to circle what specific part of the problem Ericsson wanted the project to solve.

Therefore, one of the most important milestones in the project was to identifying that the focus should be how a modular product family effects organization and process, instead of the actual product. Most research focus on modelling the product and evaluate the products attributes, e.g. production cost and performance. These aspects need different methods and tools compared to the organizational and process related aspects.

The delimitation to focus on development time and related aspects was a necessary decision due to the projects time-limit. There are though big opportunities for further research and work related to other aspects that are important in the early decomposition of the product family.

The iterative method was very successful, where sketches and prototyping was used together with validation with stakeholders. New ideas and requirements were revealed when validating early sketches and early prototypes made sure that the project was on the right track.

4.1.2. Results

The overall purpose of this project was to investigate how visualization could help an organization to understand, evaluate and communicate how different modular plans affects multiple aspects. The most important aspects for Ericsson, except overall product structure, was early identified to be resource usage, time to market and work load. Based on the feedback from stakeholders, the presented visualization and tool will very likely help Ericsson in their work.

Previous research also confirms the hypothesis that visualization is a good tool for understanding big amount of information and simplifies the communication, see section 2.3.

Validation with real data is needed to establish that the tool really works in real development. This validation is preferred to be done in everyday work with real time estimations. There is though a problem since the time span between the selection of a modular plan is years ahead of the total outcome of the developed product family. And since it is only the selected modular plan that will be implemented, there is no opportunity to verify that the best modular plan was selected. Therefore, suggested visualizations and tool should only be seen as a help in organizing and evaluating estimations made by humans and not a magic formula that output one perfect solution. But a tool with structured estimations, visualized in a good way, can help multiple stakeholders, with different viewpoint, to make better decisions. This project has gained support for that.

The web-based prototype demonstrates that web-technologies can advantageously be used for this kind of tools. The requirements that multiple users should have access and the importance of interactive graphical visualizations are well covered by the tool. No installation is required and continuous improvements and updates can easily be released. An increasing number of users and amount of data can lead to performance problems, with the consequence of longer response time.

Performance optimizations might therefore be needed in the future for good usability. Since Ericsson has a centralized system to ensure updated web-browsers the compatibility issue is not as worrying as if the tool would have been available outside the company.

Some features, implemented in the tool, could easily be extended with more types of elements to broaden the perspective from today's elements. From only representing some kind of physical objects, elements could also represent software and test-cases. Both can contain other elements, require resources, are time-planned and can be reused on different levels. This would not only give a total better understanding of the consequences of modularity, but also give better understanding of interfaces and dependencies between different organizations.

Ericsson has computer systems for time-planning, requirement handling and product structures. These are all containing data for the implemented product family and do not support any early analysis for selecting a modular plan. Integration between the suggested tool and these systems could though give big benefits. The ability to validate previous time-estimations with the real outcome is only one example.

4.2. Conclusion

Visualization plays an important role in the understanding of information and can be a useful tool when dealing with information related to product modularity. This project suggests multiple views as a support for stakeholders, with different viewpoints, in their decisions related to modular planning.

The prototype developed in this project has been validated by stakeholders within Ericsson with the feedback that it very likely would help Ericsson in their work related to modular planning. It would also very likely help them to communicate the selected modular plan more efficiently within the organization. This is giving support for the first hypothesis stated in section 1.4. Validation with real data is needed to establish the results and would preferably be done in the daily work at Ericsson.

The developed tool, using web-technologies, have shown that the desired visualizations can be developed within a web-application with great results. The main arguments are good access for multiple users together with the ability to generate great visuals without any extra installations. This is confirming the second hypothesis stated in section 1.4.

5. FUTURE WORK

The first and most important future work is to validate the prototype with real data within the pre-development at Ericsson to establish the results. This validation would preferably be done with the back-end in place and ongoing in the every-day-work.

This project has focused on how to visualize aspects related to development time, time-plan and resource usage. There are opportunities for further research to investigate how other aspects could be visualized and integrated with the views suggested in this project. Would it be possible to include estimations on costs early in the planning phase for a modular plan? Could performance requirements, e.g. power consumption, size, power dissipation, be handled? These are all aspects that could improve the decision on selecting a modular plan.

Based on the research done in this project multiple features could directly be implemented. As mentioned in the discussion, section 4.1, the type of elements can be extended to also support software and test-cases.

The tool provides multiple views and visuals that are useful in presentations and discussions, but sometimes further analyses are needed based on the data behind the visuals. Therefore a great feature would be to enable users to export specific data related to a view for analysis in Microsoft Excel or other applications.

Further investigation is suggested on the ability to integrate the suggested tool with existing systems at Ericsson. This would save time and effort by reusing data already available in other systems. Integration with the time-report system could enable evaluations of previous estimations by comparing it with the real outcome, and improve future estimations.

6. REFERENCES

AngularJS. (2014). Retrieved 2014-05-07, 2014, from https://angularjs.org/

Birst. (2014). Enterprise-caliber cloud BI | Birst. from www.birst.com

Boom, L. v. d. (2014). jQuery treetable. Retrieved 2014-05-07, 2014, from http://ludo.cubicphuse.nl/jquery-treetable/

Bostock, M. (2013). D3.js - Data-Driven Documents. Retrieved 2014-05-07, 2014, from http://d3js.org/

Bruun, H. P. L., Mortensen, N. H., & Harlou, U. (2013). Interface diagram: Design tool for supporting the development of modularity in complex product systems. *Concurrent Engineering*. doi: 10.1177/1063293x13516329

Bucket Admin. (2014). Retrieved 2014-05-07, 2014, from http://themeforest.net/item/bucket-admin-bootstrap-3-responsive-flat-dashboard/6642985

ChartJS - HTML5 JavaScript dynamic client side data visualization. (2014). Retrieved 2014-0521, 2014, from http://chartjs.devexpress.com/

Dhakar, L. (2007). Coffee Drinks Illustrated. Retrieved 2014-05-21, 2014, from http://lokeshdhakar.com/coffee-drinks-illustrated/

Eppinger, S. D. (1994). Integration analysis of product decompositions (Vol. 3690-94.): Massachusetts Institute of Technology (MIT), Sloan School of Management.

Fekete, J. (2004, 0-0 0). *The InfoVis Toolkit*. Paper presented at the Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on.

Fixson, S. K. (2003). The Mutiple Faces of Modularity - A Literature Analysis of a Product Concept for Assembled Hardware Products: University of Michigan.

Fowler, S., & Stanwick, V. (2004). *Interactive Technologies : Web Application Design Handbook : Best Practices for Web-Based Software*. Burlington, MA, USA: Morgan Kaufmann.

Harlou, U. (2006). *Developing product families based on architectures*.

Hesami, S. (2012). Modular Design and Visualization of Radio Units (pp. 60).

Holm, M. (2014). *Visualiseringsverktyg för modulär produktutveckling*. ((Bachelor Thesis), KTH, Stockholm.

JavaScript/HTML Organizational, Family, PERT & Dependencies Chart. (2014). Retrieved 2014-05-06, 2014, from http://www.basicprimitives.com/

Jiao, J., Simpson, T., & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing, 18*(1), 5-29. doi: 10.1007/s10845-007-0003-2

Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent Manufacturing, 16*(3), 371-390. doi: 10.1007/s10845-005-7030-7

Lange, M., & Imsdahl, A. (2014). Modular Function Deployment : Using Module Drivers to Impart Strategies to a Product Architecture: KTH, Tillämpad maskinteknik (KTH Södertälje).

Liu, Z., Wong, Y. S., & Lee, K. S. (2009). Modularity analysis and commonality design: a framework for the top-down platform and product family design. *International Journal of Production Research, 48*(12), 3657-3680. doi: 10.1080/00207540902902598

Metacase. (2014). MetaEdit+ Domain-Specific Modeling (DSM) environment. Retrieved 2014-05-21, 2014, from http://www.metacase.com/products.html

Microsoft. (2014a). Microsoft Office. Retrieved 2014-05-16, 2014, from http://office.microsoft.com/sv-se/

Microsoft. (2014b). Visualize with Power View and Power Map for Excel. Retrieved 2014-05-16, 2014, from http://www.microsoft.com/en-us/powerBI/home/visualize.aspx

Mikkola, J. H., & Gassmann, O. (2003). Managing modularity of product architectures: toward an integrated theory. *Engineering Management, IEEE Transactions on, 50*(2), 204-218. doi: 10.1109/TEM.2003.810826
Ohlsson, N., Norling, P., & Largren, R. (2013). *Radio System & Radio Unit HW*.
Robertson, D., & Ulrich, K. (1998). To appear in Sloan Management Review. PLATFORM PRODUCT DEVELOPMENT.
Rodden, K. (2014). Sequences sunburst. Retrieved 2014-05-07, 2014, from http://bl.ocks.org/kerryrodden/7090426
Schweighauser, M. (2014). Gantt chart for Angular.js. Retrieved 2014-05-07, 2014, from http://ngmodules.org/modules/angular-gantt
Simpson, T. W. (2004). Product platform design and customization: Status and promise. *AIEDAM, 18*(1), 3-20. doi: 10.1017/S0890060404040028
Snow, J. (1855). *On the Mode of Communication of Cholera*. Southhampton Buildings, London, England: C.F. Cheffins, Lith.
Survey: Firms Increasing Reliance on Web Application Security Testing Tools. (2013). [Press release]. Retrieved from http://www.esg-global.com/who-we-are/news-releases/survey-firms-increasing-reliance-on-web-application-security-testing-tools/
Swing Depot: Component Suites. (2004). Retrieved 2014-05-16, 2014, from http://www.javadesktop.org/rollups/components/
Telecom Training. (2007). Apis Technincal Training.
Thevenot, H., & Simpson, T. (2006). Commonality indices for product family design: a detailed comparison. *Journal of Engineering Design, 17*(2), 99-119. doi: 10.1080/09544820500275693
Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, Conn.: Cheshire, Conn. : Graphic Press.
Tufte, E. R. (1997). *Visual explanations : images and quantities, evidence and narrative*. Cheshire, Conn.: Cheshire, Conn. : Graphic Press.
Wacker, J. G., & Treleven, M. (1986). Component part standardization: An analysis of commonality sources and indices. *Journal of Operations Management, 6*(2), 219-244. doi: 10.1016/0272-6963(86)90026-4
Yau, N. (2013). *Data Points: Visualization That Means Something*: United States: John Wiley & Sons Inc.
yFiles for Java. (2014). Retrieved 2014-05-19, 2014, from http://www.yworks.com/en/products_yfiles_about.html

Functional requirements

F1) The user shall be able to select what item to base the visualization on

F2) The models used per view shall be according to Table 9.

Table 9. Views with its corresponding models

MODELS \ LEVELS	Modular Plan	Product Type	Product	Frequency Variant	Subassembly	Physical Element	Logical Element
Doughnut-chart				✓	✓	✓	
Bubble-chart		✓					
Stacked Bar-chart	✓	✓	✓	✓	✓	✓	✓
Stacked Column-chart	✓	✓	✓				
Interactive Organizational chart	✓	✓	✓	✓	✓	✓	✓
GANTT-chart	✓	✓	✓	✓	✓	✓	✓
Expandable Table	✓	✓	✓	✓	✓	✓	
Expandable Resource Table	✓	✓	✓	✓	✓	✓	
Stacked Area-chart	✓	✓	✓	✓	✓	✓	✓
Interface Matrix				✓	✓	✓	
Sunburst			✓				

F3) The user shall be able to compare items.

a) Compare shall support multiple items from multiple Modular Plans as long as they are located at the same level

b) The compare view shall contain separate models per Modular Plan displayed in the same view.

Elements

F4) All elements shall be accessible in a table.

a) The table shall be able to filter by typing a filter key per column

b) The table shall be able to sort by each column

c) The table shall contain the following columns:

i) Name

ii) Type

iii) Basic Feature

iv) IBW DL

v) IBW UL

vi) Number of Transmitters

vii) Number of Receiver

viii) Power

ix) Development Start

x) Release Date

xi) Valid To

xii) Development Time

xiii) Geometry

xiv) Used in Products

xv) Used in Elements

d) Each row shall have a link to more information

Doughnut-chart

F5) A Doughnut Chart shall support visualization of reuse

a) When hover a segment the percentage and label shall appear as a tooltip

b) Multiple doughnut charts shall be supported in one view

Bubble-chart

F6) A Bubble Chart shall support visualization of development time for multiple frequency variants of different products.

a) Frequency Bands shall be displayed as columns

b) Products shall be displayed as rows

c) Frequency variants of a product shall be represented as a bubble in the intersecting cells.

d) The size of the bubble shall be proportional to the development time.

e) The development time shall be displayed in connection to the bubble.

f) When hover a bubble, the label and amount shall be displayed as a tooltip

Stacked Bar-chart

F7) A Bar Chart shall support visualization of development time and resource usage.

a) Sections shall represent resources or containing items

b) The amount shall be displayed in connection to the section

c) When hover a section the label and amount shall be displayed as a tooltip

Stacked Column-chart

F8) A Column Chart shall support visualization of development time and resource usage.

a) Multiple columns shall be supported

b) Each column shall be able to represent a frequency variant, product, product type or modular plan

c) Sections shall represent resources or containing items

d) When hover a section the label and amount shall be displayed as a tooltip

Interactive Organizational chart

F9) An interactive organizational chart shall support visualization of product structure.

a) Each branch shall be able to display the whole structure with all levels
   i) Modularity Plan
   ii) Product Type
   iii) Product
   iv) Frequency Variant
   v) Sub-assembly
   vi) Physical Element
   vii) Logical Element
   viii) Component

b) Each branch shall, by the user, be able to show and hide to only show desired information

c) Each item shall contain a link for more information

GANTT Chart

F10) A GANTT chart shall support visualization of time plan.

a) Each item shall display development and valid period

b) If an item has children(s) the user shall be able to expand it to see internal time plan

Expandable Table

F11) An expandable table shall support visualization of contain-of relations

a) The table shall support a variable number of attributes

b) Each attribute shall be displayed in a separate column

c) Each level containing children(s) shall be expandable and collapsible by the user

d) Each row shall always have a link for more information

Expandable Resource Table

F12) An expandable Resource Table shall support visualization of resource usage amongst multiple levels.

a) Each level containing children(s) shall be expandable and collapsible by the user
b) Each row shall have a stacked bar chart displaying resource usage
   i) The requirements for F7 are applied
c) Each row shall always have a link for more information

Interface Matrix

F13) An Interactive Interface Matrix shall support visualization of interfaces

a) All elements shall be visualized both as rows and columns

b) Interfaces between two elements shall be displayed in the intersected cells

c) Elements containing children shall be able to expand to view internal interfaces

d) The Matrix shall support multiple types of interfaces

   i) Digital signal

   ii) RF

   iii) Voltage

   iv) Thermic

   v) Mechanical

e) The severity of each interface shall be categorized on a scale

   1. Minimal
   3. Easy
   5. Medium
   7. Challenging
   9. Very Complex

Stacked Area-chart

F14) A Stacked Area Chart shall support visualization of work load.

a) Each stacked area shall represent a resource or a containing item.

b) The time-scale should be able to change by the user.

c) The value of y-axis should be able to specify by the user

Sunburst

F15) A sunburst shall support visualization of shared elements between multiple frequency variants.

a) Each section shall represent a frequency variant

b) Each layer shall represent a basic feature

c) Each unique element shall have a unique color

d) A shared element shall have the same color amongst different frequency variants

e) When hover an element all entities of the element shall be highlighted.

f) When hover an element more information shall appear as a tooltip

g) Each element shall contain a link for more information

Technical Requirements

T1) The views shall be accessible for up to 300 users within Ericsson's internal network.

T2) The views shall not require any installations not provided for free Ericsson IT-service.

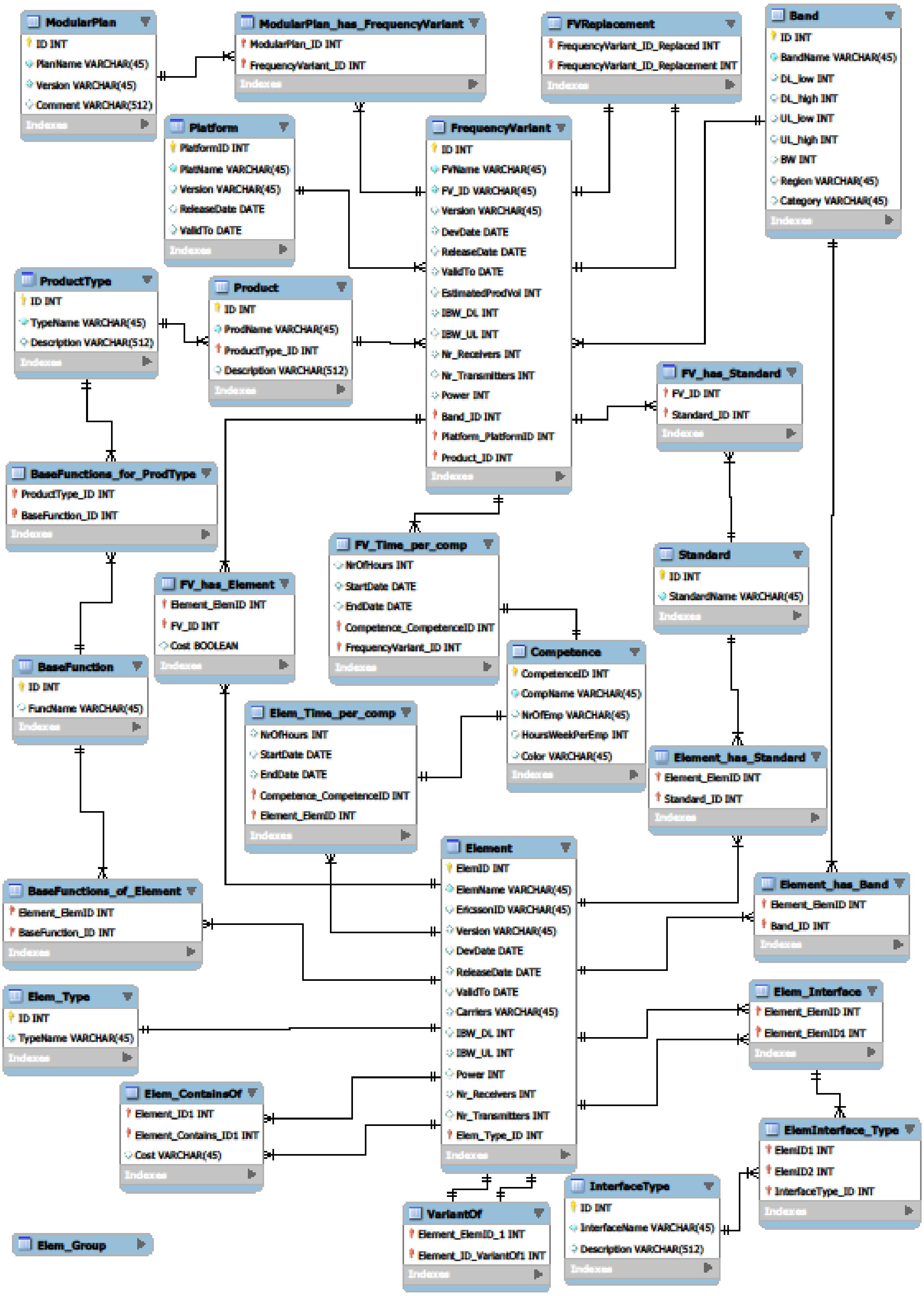
T3) The views shall be generated from a centralized set of data.

T4) The views shall be able to display on a projector.

T5) The tool shall support centralized updates without requiring any user actions.

T6) The tool shall be built on well-known technologies to facilitate future development.

APPENDIX 2 – DATA STRUCTURE



APPENDIX 3 – JSON STRUCTURE


```
/////////////////////////
//  FOR A VIEW          //
/////////////////////////

{
// GENERAL INFORMATION //
    // One of the following: Modular Plan, Product Type, Product Group, Product, Element
    "level": "Product",

    "name": "RRUS01 B0",

    // Image displayed at the top, only valid for: Product Type, Product Group & Product
    "image": "img/products/RRUS.jpg",

    // Id in our system
    "dataId": 1111,

    // Ericsson ID
    "id": "KRC 118 72",

    // Valid for, Product Group, Product & Element
    "type": {
        "name": "RRUS",
        "id": 1234
    },

    // Platform
    "platform": [
        {"name": "Platforms 4","id": 2},
        {"name": "Platforms 5","id": 3}
    ],

    // Standard
    "standard" : [
        {"name": "GSM", "id": 12},
        {"name": "LTE", "id": 13},
        {"name": "WCDMA", "id": 14}
    ],

    // Data
    "data":{
        "nrOfTransceivers": 4,
        "nrOfResceivers": 4,
        "power": 25,
        "IBWDL" : 20,
        "IBWUL" : 20
    },

    // General time-plan for item
    "planning":{
        "devStartDate": "2014-03-06",
        "releaseDate": "2014-12-06",
        "validToDate": "2015-12-06"
    },


// REUSE //
    // Subassembly reuse
    "subassemblyReuse":{
        "reuseNr": 1,
        "totalNr": 3,
        "reuseHours": 2530,
        "totalHours": 6048,
        "pei": [
            { "labelName": "Reuse", "value": 2530 },
            { "labelName": "New", "value": 3761 }
        ]
    },
    // Physical reuse
    "physicalReuse":{
        "reuseNr": 5,
        "totalNr": 9,
        "reuseHours": 2560,
        "totalHours": 4970,
        "pei": [
            { "labelName": "Reuse", "value": 2560 },
            { "labelName": "New", "value": 2410 }
        ]
    },
    // Logical reuse
    "logicalReuse":{
        "reuseNr": 2,
        "totalNr": 6,
        "reuseHours": 1504,
        "totalHours": 2199,
```

```
        "pei": [
            { "labelName": "Reuse", "value": 1504 },
            { "labelName": "New", "value": 695 }
        ]
    },


// CONTAIN-OF TABLE //
// Table with all "containing X" and information about them.
//      For Product: only elements
//      For Product Group: Products and their elements
//      For Product Type: Product Groups, their products and elements
// In "labels" all the rows needs to be defined. The data needs to contain values for all specified values.

"contain":{
    "labels": [{"value":"Name"}, {"value":"Type"}, {"value":"Platform"}],
    "data":[
        {"id": 10001, "data": [{"value": "Subass 1"}, {"value": "attr 2"}, {"value": "Platform X"}],"children":[
            {"id": 100011, "data": [{"value": "Physical Element 1"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100012, "data": [{"value": "Physical Element 2"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100013, "data": [{"value": "Physical Element 3"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100014, "data": [{"value": "Physical Element 4"}, {"value": "attr 2"}, {"value": "Platform X"}]}
        ]},
        {"id": 2, "data": [{"value": "Subass 2"}, {"value": "attr 2"}, {"value": "Platform X"}], "children":[
            {"id": 100021, "data": [{"value": "Physical Element 1"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100022, "data": [{"value": "Physical Element 2"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100023, "data": [{"value": "Physical Element 3"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100024, "data": [{"value": "Physical Element 4"}, {"value": "attr 2"}, {"value": "Platform X"}]}
        ]},
        {"id": 10003, "data": [{"value": "Subass 3"}, {"value": "attr 2"}, {"value": "Platform X"}], "children":[
            {"id": 100031, "data": [{"value": "Physical Element 1"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100032, "data": [{"value": "Physical Element 2"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100033, "data": [{"value": "Physical Element 3"}, {"value": "attr 2"}, {"value": "Platform X"}]},
            {"id": 100034, "data": [{"value": "Physical Element 4"}, {"value": "attr 2"}, {"value": "Platform X"}], "children": [
                {"id": 1000341, "data": [{"value": "Logical Element 1"}, {"value": "attr 2"}, {"value": "Platform X"}], "children"
                : [
                    {"id": 10003411, "data": [{"value": "Component 1"}, {"value": "attr 2"}, {"value": "Platform X"}]},
                    {"id": 10003412, "data": [{"value": "Component 2"}, {"value": "attr 2"}, {"value": "Platform X"}]}
                ]},
                {"id": 1000342, "data": [{"value": "Logical Element 2"}, {"value": "attr 2"}, {"value": "Platform X"}]},
                {"id": 1000343, "data": [{"value": "Logical Element 3"}, {"value": "attr 2"}, {"value": "Platform X"}]}
            ]}
        ]},
        {"id": 10004, "data": [{"value": "Component 1"}, {"value": "attr 2"}, {"value": "Platform X"}]}
    ]
},


// TREVIEW //
"treeView":
[
    // Modular Plan
    {
        "id": 0,
        "parent": null,
        "title": "Modular Plan 1",
        "type": "modularPlan",
        "grouptitle": "Modular Plan"
    },

    // Type
    {
        "id": 10,
        "parent": 0,
        "title": "RRUS",
        "type": "type",
        "grouptitle": "Type",
        // Don't change
        "myItemType": "primitives.orgdiagram.ItemType.GeneralPartner",
        "myAdviserPlacementType": "primitives.orgdiagram.AdviserPlacementType.Right"
    },

    // Platform
    {
        "id": 12,
        "parent": 0,
        "title": "PL4",
        "type": "platform",
        "grouptitle": "Platform",
        // Don't change
        "myItemType": "primitives.orgdiagram.ItemType.GeneralPartner",
        "myAdviserPlacementType": "primitives.orgdiagram.AdviserPlacementType.Right"
    },

    // Band
    {
        "id": 13,
        "parent": 0,
        "title": "B20",
        "type": "band",
        "grouptitle": "Band",
        // Don't change
        "myItemType": "primitives.orgdiagram.ItemType.GeneralPartner",
        "myAdviserPlacementType": "primitives.orgdiagram.AdviserPlacementType.Right"
    },

    // Product
    {
        "id": 1111,
        "parent": 0,
        "title": "RRUS 01 - B20",
        "type": "currentProduct",
        "grouptitle": "Product"
    },
```

```
// Previous Product
{
    "id": 3,
    "parent": 1111,
    "title": "RRUS 01-OLD - B20",
    "type": "previous",
    "grouptitle": "Previous",
    "myItemType": "primitives.orgdiagram.ItemType.Adviser",
    "myAdviserPlacementType": "primitives.orgdiagram.AdviserPlacementType.Left"
},

// Replacement Product
{
    "id": 4,
    "parent": 1111,
    "title": "RRUS 02 - B20",
    "type": "replacement",
    "grouptitle": "Replacement",
    "myItemType": "primitives.orgdiagram.ItemType.Adviser",
    "myAdviserPlacementType": "primitives.orgdiagram.AdviserPlacementType.Right"
},

// Containing Elements
{
    "id": 5,
    "parent": 1111,
    "title": "SubAss 1",
    "type": "subassembly",
    "grouptitle": "Sub-Assembly"
},
{
    "id": 51,
    "parent": 5,
    "title": "Physical Element 1",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 52,
    "parent": 5,
    "title": "Physical Element 2",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 53,
    "parent": 5,
    "title": "Physical Element 3",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 54,
    "parent": 5,
    "title": "Physical Element 4",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 6,
    "parent": 1111,
    "title": "SubAss 2",
    "type": "subassembly",
    "grouptitle": "Sub-Assembly"
},
{
    "id": 61,
    "parent": 6,
    "title": "Physical Element 1",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 62,
    "parent": 6,
    "title": "Physical Element 2",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 63,
    "parent": 6,
    "title": "Physical Element 3",
    "type": "physical",
    "grouptitle": "Physical Element"
},
{
    "id": 64,
    "parent": 6,
    "title": "Physical Element 4",
    "type": "physical",
    "grouptitle": "Physical Element"
},

{
    "id": 7,
    "parent": 1111,
    "title": "Component 1",
    "type": "component",
    "grouptitle": "Component"
},
{
    "id": 8,
```

```
        "parent": 1111,
        "title": "SubAss 3",
        "type": "subassembly",
        "grouptitle": "Sub-Assembly"
    },
    {
        "id": 81,
        "parent": 8,
        "title": "Physical Element 1",
        "type": "physical",
        "grouptitle": "Physical Element"
    },
    {
        "id": 82,
        "parent": 8,
        "title": "Physical Element 2",
        "type": "physical",
        "grouptitle": "Physical Element"
    },
    {
        "id": 83,
        "parent": 8,
        "title": "Physical Element 3",
        "type": "physical",
        "grouptitle": "Physical Element"
    },
    {
        "id": 84,
        "parent": 8,
        "title": "Physical Element 4",
        "type": "physical",
        "grouptitle": "Physical Element"
    },
    {
        "id": 841,
        "parent": 84,
        "title": "Locial Element 1",
        "type": "logical",
        "grouptitle": "Locial Element"
    },
    {
        "id": 842,
        "parent": 84,
        "title": "Locial Element 2",
        "type": "logical",
        "grouptitle": "Locial Element"
    },
    {
        "id": 843,
        "parent": 84,
        "title": "Locial Element 3",
        "type": "logical",
        "grouptitle": "Locial Element"
    },
    {
        "id": 8411,
        "parent": 841,
        "title": "Component 1",
        "type": "component",
        "grouptitle": "Component"
    },
    {
        "id": 8412,
        "parent": 841,
        "title": "Component 2",
        "type": "component",
        "grouptitle": "Component"
    }
],

// RESOURCE USAGE - per resource //
    "resourceUsageLabels":[
        { "valueField": "Resource 1", "name": "Resource 1"},
        { "valueField": "Resource 2", "name": "Resource 2"},
        { "valueField": "Resource 3", "name": "Resource 2"},
        { "valueField": "Resource 4", "name": "Resource 4"},
        { "valueField": "Resource 5", "name": "Resource 5"}
    ],
    "resourceUsage":[
        { "label": "",
        "Resource 1": 5000,
        "Resource 2": 10000,
        "Resource 3": 25000,
        "Resource 4": 10000,
        "Resource 5": 25000
        }
    ],

// RESOURCE USAGE - per item //
    "resourceUsageProductLabels":[
        { "valueField": "Subassembly C - 1", "name": "Subassembly C - 1"},
        { "valueField": "Subassembly B2 - 2", "name": "Subassembly B2 - 2"},
        { "valueField": "Subassembly B2 - 3", "name": "Subassembly B2 - 3"}
    ],

    "resourceUsageProduct":[
        { "label": "",
        "Subassembly C - 1": 45000,
        "Subassembly B2 - 2": 37000,
        "Subassembly B2 - 3": 38000
        }
    ],
```

```
// TIME PLAN //
    "timePlan":[
        {"id": 2112, "description": "RRUS01 B2", "order": 1, "level": 0, "tasks": [
            {"id": "dev1111", "subject": "RRUS01 B2 - Development", "color": "#6f2664", "from": "2014-08-01T00:00:00", "to":
            "2015-03-30T00:00:00"},
            {"id": "rel1111", "subject": "RRUS01 B2 - 1 - Available", "color": "#90b9b6", "from": "2015-03-30T00:00:00", "to":
            "2016-06-01T00:00:00"}
        ], "data":{"children":[
            {"id": 21111, "description": "Subassembly C - 1", "level": 1, "order": 2, "tasks": [
                {"id": "dev21111", "subject": "Subassembly C - 1 - Development", "color": "#6f2664", "from": "2014-01-01T00:00:00"
                , "to": "2014-06-30T00:00:00"},
                {"id": "rel21111", "subject": "Subassembly C - 1 - Available", "color": "#90b9b6", "from": "2014-06-30T00:00:00",
                "to": "2016-06-14T00:00:00"}
            ], "data":{ "children":[
                {"id": 211111, "description": "Physical Element C - 1.1", "level": 2, "order": 3, "tasks": [
                    {"id": "dev211111", "subject": "Physical Element C - 1.1 - Development", "color": "#6f2664", "from":
                    "2014-01-01T00:00:00", "to": "2014-04-30T00:00:00"},
                    {"id": "rel211111", "subject": "Physical Element C - 1.1 - Available", "color": "#90b9b6", "from":
                    "2014-04-30T00:00:00", "to": "2016-06-14T00:00:00"}
                ], "data":{"children":[
                        {"id": 2111111, "description": "Logical Element C - 1.1.1", "level": 3, "order": 4, "tasks": [
                            {"id": "dev211112", "subject": "Logical Element C - 1.1.1 - Development", "color": "#844b7c", "from":
                            "2014-01-01T00:00:00", "to": "2014-02-15T00:00:00"},
                            {"id": "rel211112", "subject": "Logical Element C - 1.1.1 - Available", "color": "#007b78", "from":
                            "2014-02-15T00:00:00", "to": "2016-06-14T00:00:00"}
                        ]},
                        {"id": 2111112, "description": "Logical Element C - 1.1.2", "level": 3, "order": 5, "tasks": [
                            {"id": "dev2111112", "subject": "Logical Element C - 1.1.2 - Development", "color": "#844b7c", "from":
                             "2014-01-01T00:00:00", "to": "2014-02-10T00:00:00"},
                            {"id": "rel2111112", "subject": "Logical Element C - 1.1.2 - Available", "color": "#007b78", "from":
                            "2014-02-10T00:00:00", "to": "2016-06-14T00:00:00"}
                        ]}
                    ]}
                },
                {"id": 211112, "description": "Physical Element C - 1.2", "level": 2, "order": 6, "tasks": [
                    {"id": "dev211112", "subject": "Physical Element B2 - 1.2 - Development", "color": "#6f2664", "from":
                    "2014-01-01T00:00:00", "to": "2014-04-28T00:00:00"},
                    {"id": "rel211112", "subject": "Physical Element B2 - 1.2 - Available", "color": "#90b9b6", "from":
                    "2014-02-28T00:00:00", "to": "2016-06-14T00:00:00"}
                ]},
                {"id": 211113, "description": "Physical Element C - 1.3", "level": 2, "order": 6.5, "tasks": [
                    {"id": "dev211113", "subject": "Physical Element B2 - 1.3 - Development", "color": "#6f2664", "from":
                    "2014-01-01T00:00:00", "to": "2014-04-28T00:00:00"},
                    {"id": "rel211113", "subject": "Physical Element B2 - 1.3 - Available", "color": "#90b9b6", "from":
                    "2014-02-28T00:00:00", "to": "2016-06-14T00:00:00"}
                ]}
            ]}},
        ]}}
    ],


// WORK LOAD - Per Resource //
    // Labels
    "workLoadLabels":[
        {
            "name": "Resource1",
            "valueField": "Resource1"
        },{
            "name": "Resource2",
            "valueField": "Resource2"
        },{
            "name": "Resource3",
            "valueField": "Resource3"
        },{
            "name": "Resource4",
            "valueField": "Resource4"
        },{
            "name": "Resource5",
            "valueField": "Resource5"
        }
    ],

    // Data
    // Two new object are needed every time any resource change. One for the previous value and one for the new.
    // These two datapoints should be defined with only one second difference to simulate discrete graph.
    "workLoad":[
        { "date": "2013-01-01", "Resource2": 227, "Resource3": 331, "Resource4": 1436, "Resource5": 547, "Resource1": 120},
        { "date": "2013-01-15T00:00:00", "Resource2": 227, "Resource3": 331, "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        { "date": "2013-01-15T00:00:01", "Resource2": 750, "Resource3": 331, "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        { "date": "2013-02-01T00:00:00", "Resource2": 750, "Resource3": 331, "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        { "date": "2013-02-01T00:00:01", "Resource2": 285, "Resource3": 416, "Resource4": 1718, "Resource5": 605, "Resource1": 150}
    ],


// WORK LOAD - per item //
// Defined as "per resource" but with different labels.
    "workLoadProductLabels":[
        {
            "name": "RRUS01 B0",
            "valueField": "RRUS01 B0"
        },{
            "name": "RRUS01 B1",
            "valueField": "RRUS01 B1"
        },{
            "name": "RRUS01 B2",
            "valueField": "RRUS01 B2"
        },{
            "name": "RRUS01 B3",
```

```
/////////////////
// COMPARE      //
/////////////////
// In general is compare defined in the same way as the regular view.
// Examples where things that are different are shown in this example

{
// OBJECTS TO COMPARE //
    "comparedObjects": [
        {"id": 1, "name": "Modular Plan 1", "objects":[
            {"id": "111", "type": "Product", "name": "RRUS01"}
        ]},
        {"id": 2, "name": "Modular Plan 2", "objects":[
            {"id": "211", "type": "Product", "name": "RRUS01"}
        ]}
    ],


// RESOURCE USAGE - per resource //
    "resourceUsageLabels":[
        { "valueField": "Resource 1", "name": "Resource 1"},
        { "valueField": "Resource 2", "name": "Resource 2"},
        { "valueField": "Resource 3", "name": "Resource 2"},
        { "valueField": "Resource 4", "name": "Resource 4"},
        { "valueField": "Resource 5", "name": "Resource 5"}
    ],
    "resourceUsage":[
        { "label": "Modular Plan 2",
        "Resource 1": 2372,
        "Resource 2": 2068,
        "Resource 3": 2544,
        "Resource 4": 1405,
        "Resource 5": 1584
        },
        { "label": "Modular Plan 1",
        "Resource 1": 2502,
        "Resource 2": 2639,
        "Resource 3": 2762,
        "Resource 4": 1485,
        "Resource 5": 1762
        }
    ],


// RESOURCE USAGE - per item //
"resourceUsageProductLabels":[
    { "valueField": "RRUS01 B0", "name": "RRUS01 B0"},
    { "valueField": "RRUS01 B1", "name": "RRUS01 B1"},
    { "valueField": "RRUS01 B2", "name": "RRUS01 B2"},
    { "valueField": "RRUS01 B3", "name": "RRUS01 B3"},
    { "valueField": "RRUS01 B5", "name": "RRUS01 B5"},
    { "valueField": "RRUS01 B8", "name": "RRUS01 B8"},
    { "valueField": "RRUS01 B9", "name": "RRUS01 B9"},
    { "valueField": "RRUS01 B14", "name": "RRUS01 B14"},
    { "valueField": "RRUS01 B20", "name": "RRUS01 B20"}
],
"resourceUsageProduct":[
    { "label": "Modular Plan 2",
    "RRUS01 B0": 6291,
    "RRUS01 B1": 1582,
    "RRUS01 B2": 2100
    },
    { "label": "Modular Plan 1",
    "RRUS01 B0": 4558,
    "RRUS01 B1": 3564,
    "RRUS01 B2": 3028
    }
],



// RESOURCE USAGE - Grouped per frequency variant //
"resourceProductSeries": [
    { "valueField": "resource 1.1", "name": "M1 - Resource 1", "stack": "Modular Plan 1" },
    { "valueField": "resource 1.2", "name": "M1 - Resource 2", "stack": "Modular Plan 1" },
    { "valueField": "resource 1.3", "name": "M1 - Resource 3", "stack": "Modular Plan 1" },
    { "valueField": "resource 1.4", "name": "M1 - Resource 4", "stack": "Modular Plan 1" },
    { "valueField": "resource 1.5", "name": "M1 - Resource 5", "stack": "Modular Plan 1" },
    { "valueField": "resource 2.1", "name": "M2 - Resource 1", "stack": "Modular Plan 2" },
    { "valueField": "resource 2.2", "name": "M2 - Resource 2", "stack": "Modular Plan 2" },
    { "valueField": "resource 2.3", "name": "M2 - Resource 3", "stack": "Modular Plan 2" },
    { "valueField": "resource 2.4", "name": "M2 - Resource 4", "stack": "Modular Plan 2" },
    { "valueField": "resource 2.5", "name": "M2 - Resource 5", "stack": "Modular Plan 2" }
],
"resourceProduct": [
{ "product": "RRUS01 B0",
    "resource 1.1": 972, "resource 1.2": 1152, "resource 1.3": 1173, "resource 1.4": 577, "resource 1.5": 684,
    "resource 2.1": 1456, "resource 2.2": 1337, "resource 2.3": 1720, "resource 2.4": 890, "resource 2.5": 888
},
{ "product": "RRUS01 B1",
    "resource 1.1": 828, "resource 1.2": 804, "resource 1.3": 859, "resource 1.4": 490, "resource 1.5": 583 ,
    "resource 2.1": 576, "resource 2.2": 342, "resource 2.3": 352, "resource 2.4": 215, "resource 2.5": 340
},
{ "product": "RRUS01 B2",
    "resource 1.1": 702, "resource 1.2": 683, "resource 1.3": 730, "resource 1.4": 418, "resource 1.5": 495,
    "resource 2.1": 431, "resource 2.2": 403, "resource 2.3": 522, "resource 2.4": 316, "resource 2.5": 428
},
{ "product": "RRUS01 B3",
    "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
    "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
},
{ "product": "RRUS01 B5",
    "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
```

```
        "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
    },
    { "product": "RRUS01 B8",
        "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
        "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
    },
    { "product": "RRUS01 B9",
        "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
        "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
    },
    { "product": "RRUS01 B14",
        "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
        "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
    },
    { "product": "RRUS01 B20",
        "resource 1.1": 0, "resource 1.2": 0, "resource 1.3": 0, "resource 1.4": 0, "resource 1.5": 0,
        "resource 2.1": 0, "resource 2.2": 0, "resource 2.3": 0, "resource 2.4": 0, "resource 2.5": 0
    }
],


    "timePlan":[

        {"id": 1, "description": "ModularPlan 1", "order": 0, "tasks": [
            {"id": "dev1", "subject": "ModularPlan 1 - Development", "color": "#6f2664", "from": "2014-01-01T00:00:00", "to":
            "2014-08-01T00:00:00"},
            {"id": "rel1", "subject": "ModularPlan 1 - Available", "color": "#00625f", "from": "2014-08-01T00:00:00", "to":
            "2016-06-01T00:00:00"}
        ], "data":{"level": 0, "children":[
            {"id": 1111, "description": "RRUS01 B0", "order": 1, "tasks": [
                {"id": "dev1111", "subject": "RRUS01 B0 - Development", "color": "#6f2664", "from": "2014-01-01T00:00:00", "to":
                "2014-06-01T00:00:00"},
                {"id": "rel1111", "subject": "RRUS01 B0 - 1 - Available", "color": "#00625f", "from": "2014-06-01T00:00:00", "to":
                "2016-06-01T00:00:00"}
            ], "data":{"level": 1, "children":[
                {"id": 11111, "description": "Subassembly B0 - 1", "order": 2, "tasks": [
                    {"id": "dev11111", "subject": "Subassembly B0 - 1 - Development", "color": "#6f2664", "from":
                    "2014-01-01T00:00:00", "to": "2014-04-30T00:00:00"},
                    {"id": "rel11111", "subject": "Subassembly B0 - 1 - Available", "color": "#00625f", "from": "2014-04-30T00:00:00",
                     "to": "2016-06-01T00:00:00"}
                ], "data":{"level": 2, "children":[
                    {"id": 111111, "description": "Physical Element B0 - 1.1", "order": 3, "tasks": [
                        {"id": "dev111111", "subject": "Physical Element B0 - 1.1 - Development", "color": "#6f2664", "from":
                        "2014-01-01T00:00:00", "to": "2014-03-10T00:00:00"},
                        {"id": "rel111111", "subject": "Physical Element B0 - 1.1 - Available", "color": "#00625f", "from":
                        "2014-03-10T00:00:00", "to": "2016-06-03T00:00:00"}
                    ], "data":{"level": 3, "children":[
                            {"id": 1111111, "description": "Logical Element B0 - 1.1.1", "order": 4, "tasks": [
                                {"id": "dev111111", "subject": "Logical Element B0 - 1.1.1 - Development", "color": "#844b7c", "from":
                                 "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                                {"id": "rel111111", "subject": "Logical Element B0 - 1.1.1 - Available", "color": "#007b78", "from":
                                "2014-01-30T00:00:00", "to": "2016-06-03T00:00:00"}
                            ], "data":{"level": 4}},
                            {"id": 1111112, "description": "Logical Element B0 - 1.1.2", "order": 5, "tasks": [
                                {"id": "dev1111112", "subject": "Logical Element B0 - 1.1.2 - Development", "color": "#844b7c", "from"
                                : "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                                {"id": "rel1111112", "subject": "Logical Element B0 - 1.1.2 - Available", "color": "#007b78", "from":
                                "2014-01-30T00:00:00", "to": "2016-06-04T00:00:00"}
                            ], "data":{"level": 4}}
                        ]}
                    },
                    {"id": 111112, "description": "Physical Element B0 - 1.2", "order": 6, "tasks": [
                        {"id": "dev111112", "subject": "Physical Element B0 - 1.2 - Development", "color": "#6f2664", "from":
                        "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
                        {"id": "rel111112", "subject": "Physical Element B0 - 1.2 - Available", "color": "#00625f", "from":
                        "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
                    ], "data":{"level": 3}},
                    {"id": 111113, "description": "Physical Element B0 - 1.3", "order": 6, "tasks": [
                        {"id": "dev111113", "subject": "Physical Element B0 - 1.3 - Development", "color": "#6f2664", "from":
                        "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
                        {"id": "rel111113", "subject": "Physical Element B0 - 1.3 - Available", "color": "#00625f", "from":
                        "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
                    ], "data":{"level": 3}}
                ]}},
                {"id": 11112, "description": "Subassembly B0 - 2", "order": 7, "tasks": [
                    {"id": "dev11112", "subject": "Subassembly B0 - 2 - Development", "color": "#6f2664", "from":
                    "2014-01-01T00:00:00", "to": "2014-04-30T00:00:00"},
                    {"id": "rel11112", "subject": "Subassembly B0 - 2 - Available", "color": "#00625f", "from": "2014-04-30T00:00:00",
                     "to": "2016-06-01T00:00:00"}
                ], "data":{"level": 2, "children":[
                    {"id": 111121, "description": "Physical Element B0 - 2.1", "order": 8, "tasks": [
                        {"id": "dev111121", "subject": "Physical Element B0 - 2.1 - Development", "color": "#6f2664", "from":
                        "2014-01-01T00:00:00", "to": "2014-03-10T00:00:00"},
                        {"id": "rel111121", "subject": "Physical Element B0 - 2.1 - Available", "color": "#00625f", "from":
                        "2014-03-10T00:00:00", "to": "2016-06-03T00:00:00"}
                    ], "data":{"level": 3, "children":[
                            {"id": 1111211, "description": "Logical Element B0 - 2.1.1", "order": 9, "tasks": [
                                {"id": "dev111211", "subject": "Logical Element B0 - 2.1.1 - Development", "color": "#844b7c", "from":
                                 "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                                {"id": "rel111211", "subject": "Logical Element B0 - 2.1.1 - Available", "color": "#007b78", "from":
                                "2014-01-30T00:00:00", "to": "2016-06-03T00:00:00"}
                            ], "data":{"level": 4}},
                            {"id": 1111212, "description": "Logical Element B0 - 2.1.2", "order": 10, "tasks": [
                                {"id": "dev1111212", "subject": "Logical Element B0 - 2.1.2 - Development", "color": "#844b7c", "from"
                                : "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                                {"id": "rel1111212", "subject": "Logical Element B0 - 2.1.2 - Available", "color": "#007b78", "from":
                                "2014-01-30T00:00:00", "to": "2016-06-04T00:00:00"}
                            ], "data":{"level": 4}}
                        ]}
                    },
                    {"id": 111122, "description": "Physical Element B0 - 2.2", "order": 11, "tasks": [
                        {"id": "dev111122", "subject": "Physical Element B0 - 2.2 - Development", "color": "#6f2664", "from":
                        "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
```

```
                {"id": "rel111122", "subject": "Physical Element B0 - 2.2 - Available", "color": "#00625f", "from":
                "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
            ], "data":{"level": 3}},
            {"id": 111123, "description": "Physical Element B0 - 2.3", "order": 12, "tasks": [
                {"id": "dev111123", "subject": "Physical Element B0 - 2.3 - Development", "color": "#6f2664", "from":
                "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
                {"id": "rel111123", "subject": "Physical Element B0 - 2.3 - Available", "color": "#00625f", "from":
                "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
            ], "data":{"level": 3}}
        ]}},
        {"id": 11113, "description": "Subassembly B0 - 3", "order": 13, "tasks": [
            {"id": "dev11113", "subject": "Subassembly B0 - 3 - Development", "color": "#6f2664", "from":
            "2014-01-01T00:00:00", "to": "2014-04-30T00:00:00"},
            {"id": "rel11113", "subject": "Subassembly B0 - 3 - Available", "color": "#00625f", "from": "2014-04-30T00:00:00",
             "to": "2016-06-01T00:00:00"}
        ], "data":{"level": 2, "children":[
            {"id": 111131, "description": "Physical Element B0 - 3.1", "order": 14, "tasks": [
                {"id": "dev111131", "subject": "Physical Element B0 - 3.1 - Development", "color": "#6f2664", "from":
                "2014-01-01T00:00:00", "to": "2014-03-10T00:00:00"},
                {"id": "rel111131", "subject": "Physical Element B0 - 3.1 - Available", "color": "#00625f", "from":
                "2014-03-10T00:00:00", "to": "2016-06-03T00:00:00"}
            ], "data":{"level": 3, "children":[
                    {"id": 1111311, "description": "Logical Element B0 - 3.1.1", "order": 15, "tasks": [
                        {"id": "dev111311", "subject": "Logical Element B0 - 3.1.1 - Development", "color": "#844b7c", "from":
                         "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                        {"id": "rel111311", "subject": "Logical Element B0 - 3.1.1 - Available", "color": "#007b78", "from":
                        "2014-01-30T00:00:00", "to": "2016-06-03T00:00:00"}
                    ], "data":{"level": 4}},
                    {"id": 1111312, "description": "Logical Element B0 - 3.1.2", "order": 16, "tasks": [
                        {"id": "dev1111312", "subject": "Logical Element B0 - 3.1.2 - Development", "color": "#844b7c", "from"
                        : "2014-01-01T00:00:00", "to": "2014-01-30T00:00:00"},
                        {"id": "rel1111312", "subject": "Logical Element B0 - 3.1.2 - Available", "color": "#007b78", "from":
                        "2014-01-30T00:00:00", "to": "2016-06-04T00:00:00"}
                    ], "data":{"level": 4}}
                ]}
            },
            {"id": 111132, "description": "Physical Element B0 - 3.2", "order": 17, "tasks": [
                {"id": "dev111132", "subject": "Physical Element B0 - 3.2 - Development", "color": "#6f2664", "from":
                "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
                {"id": "rel111132", "subject": "Physical Element B0 - 3.2 - Available", "color": "#00625f", "from":
                "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
            ], "data":{"level": 3}},
            {"id": 111133, "description": "Physical Element B0 - 3.3", "order": 18, "tasks": [
                {"id": "dev111133", "subject": "Physical Element B0 - 3.3 - Development", "color": "#6f2664", "from":
                "2014-01-01T00:00:00", "to": "2014-02-28T00:00:00"},
                {"id": "rel111133", "subject": "Physical Element B0 - 3.3 - Available", "color": "#00625f", "from":
                "2014-02-28T00:00:00", "to": "2016-06-03T00:00:00"}
            ], "data":{"level": 3}}
        ]}}
    ]}},
    {"id": 1112, "description": "RRUS01 B1", "order": 20, "tasks": [
        {"id": "dev1112", "subject": "RRUS01 B1 - Development", "color": "#6f2664", "from": "2014-02-01T00:00:00", "to":
        "2014-07-01T00:00:00"},
        {"id": "rel1112", "subject": "RRUS01 B1 - Available", "color": "#00625f", "from": "2014-07-01T00:00:00", "to":
        "2016-06-01T00:00:00"}
    ], "data":{"level": 1, "children":[{}]}},
    {"id": 1113, "description": "RRUS01 B2", "order": 22, "tasks": [
        {"id": "dev1113", "subject": "RRUS01 B2 - Development", "color": "#6f2664", "from": "2014-03-01T00:00:00", "to":
        "2014-08-01T00:00:00"},
        {"id": "rel1113", "subject": "RRUS01 B2 - Available", "color": "#00625f", "from": "2014-08-01T00:00:00", "to":
        "2016-06-01T00:00:00"}
    ], "data":{"level": 1, "children":[{}]}},
    {"id": 1114, "description": "RRUS01 B3", "order": 23, "tasks": [], "data":{"level": 1}},
    {"id": 1115, "description": "RRUS01 B5", "order": 24, "tasks": [], "data":{"level": 1}},
    {"id": 1116, "description": "RRUS01 B8", "order": 25, "tasks": [], "data":{"level": 1}},
    {"id": 1117, "description": "RRUS01 B9", "order": 26, "tasks": [], "data":{"level": 1}},
    {"id": 1118, "description": "RRUS01 B14", "order": 27, "tasks": [], "data":{"level": 1}},
    {"id": 1119, "description": "RRUS01 B20", "order": 28, "tasks": [], "data":{"level": 1}}
]}},

{"id": 2, "description": "ModularPlan 2", "order": 10000, "tasks": [
    {"id": "dev2", "subject": "ModularPlan 2 - Development", "color": "#6f2664", "from": "2014-01-01T00:00:00", "to":
    "2015-03-01T00:00:00"},
    {"id": "rel2", "subject": "ModularPlan 2 - Available", "color": "#00625f", "from": "2015-03-01T00:00:00", "to":
    "2016-06-01T00:00:00"}
], "data":{"level": 0, "children":[
    {"id": 2111, "description": "RRUS01 B0", "order": 10001, "tasks": [
        {"id": "dev1111", "subject": "RRUS01 B0 - Development", "color": "#6f2664", "from": "2014-01-01T00:00:00", "to":
        "2014-08-01T00:00:00"},
        {"id": "rel1111", "subject": "RRUS01 B0 - 1 - Available", "color": "#00625f", "from": "2014-08-01T00:00:00", "to":
        "2016-06-01T00:00:00"}
    ], "data":{"level": 1, "children":[{}]}},
    {"id": 2112, "description": "RRUS01 B1", "order": 100020, "tasks": [
        {"id": "dev1112", "subject": "RRUS01 B1 - Development", "color": "#6f2664", "from": "2014-06-01T00:00:00", "to":
        "2015-01-01T00:00:00"},
        {"id": "rel1112", "subject": "RRUS01 B1 - Available", "color": "#00625f", "from": "2015-01-01T00:00:00", "to":
        "2016-06-01T00:00:00"}
    ], "data":{"level": 1, "children":[{}]}},
    {"id": 2113, "description": "RRUS01 B2", "order": 100022, "tasks": [
        {"id": "dev1113", "subject": "RRUS01 B2 - Development", "color": "#6f2664", "from": "2014-08-01T00:00:00", "to":
        "2015-03-01T00:00:00"},
        {"id": "rel1113", "subject": "RRUS01 B2 - Available", "color": "#00625f", "from": "2015-03-01T00:00:00", "to":
        "2016-06-01T00:00:00"}
    ], "data":{"level": 1, "children":[{}]}},
    {"id": 2114, "description": "RRUS01 B3", "order": 100023, "tasks": [], "data":{"level": 1}},
    {"id": 2115, "description": "RRUS01 B5", "order": 100024, "tasks": [], "data":{"level": 1}},
    {"id": 2116, "description": "RRUS01 B8", "order": 100025, "tasks": [], "data":{"level": 1}},
    {"id": 2117, "description": "RRUS01 B9", "order": 100026, "tasks": [], "data":{"level": 1}},
    {"id": 2118, "description": "RRUS01 B14", "order": 100027, "tasks": [], "data":{"level": 1}},
    {"id": 2119, "description": "RRUS01 B20", "order": 100028, "tasks": [], "data":{"level": 1}}
]}}
],

// WORK LOAD
```

```
"workLoadLabels":[
    {
        "name": "Resource1",
        "valueField": "Resource1"
    },{
        "name": "Resource2",
        "valueField": "Resource2"
    },{
        "name": "Resource3",
        "valueField": "Resource3"
    },{
        "name": "Resource4",
        "valueField": "Resource4"
    },{
        "name": "Resource5",
        "valueField": "Resource5"
    }
],

"workLoad":[
{
    "data":[
        {
            "date": "2013-01-01T00:00:00", "Resource2": 0, "Resource3": 0,
            "Resource4": 0, "Resource5": 0, "Resource1": 0
        },
        {
            "date": "2013-01-15T00:00:00", "Resource2": 0, "Resource3": 0,
            "Resource4": 0, "Resource5": 0, "Resource1": 0
        },
        {
            "date": "2013-01-15T00:00:01", "Resource2": 750, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        {
            "date": "2013-02-01T00:00:00", "Resource2": 750, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        }
    ]
},
{
    "data":[
        {
            "date": "2013-01-01T00:00:00", "Resource2": 0, "Resource3": 0,
            "Resource4": 0, "Resource5": 0, "Resource1": 0
        },
        {
            "date": "2013-01-01T00:00:01", "Resource2": 2027, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        {
            "date": "2013-01-15T00:00:00", "Resource2": 2027, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        {
            "date": "2013-01-15T00:00:01", "Resource2": 750, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        },
        {
            "date": "2013-02-01T00:00:00", "Resource2": 750, "Resource3": 331,
            "Resource4": 1436, "Resource5": 547, "Resource1": 120
        }
    ]
}
],

"workLoadZoom":[
    {
        "date": "2013-01-01T00:00:00", "Resource2": 0, "Resource3": 0,
        "Resource4": 0, "Resource5": 0, "Resource1": 0
    },
    {
        "date": "2013-08-01T00:00:01", "Resource2": 0, "Resource3": 0,
        "Resource4": 0, "Resource5": 0, "Resource1": 0
    }
],

// SUNBURST //
"sunburst":[
            {
            "name": "",
            "children": [
                {
                "id": "class211113",
                "url": "/index.html#/components/11113",
                "name": "Subassembly B0 - 3",
                "color": "#fcd2a2",
                "children": [{
                    "id": "class211112",
                    "url": "/index.html#/components/11112",
                    "name": "Subassembly B0 - 2",
                    "color": "#f7ba98",
                    "children": [{
                        "id": "class2111111",
                        "url": "/index.html#/components/111111",
                        "name": "Physical Element B0 - 1.1",
                        "color": "#d5e4af",
                        "children": [{
                            "id": "class211111",
                            "url": "/index.html#/components/11111",
                            "name": "Subassembly B0 - 1",
                            "color": "#c19cbb",
                            "children": [{
                                "url": "/index.html#/products/1111",
```

```
                    "id": "class21111",
                    "name": "RRUS01 B0",
                    "color": "#F1F2F7",
                    "size": 10
                }]
                }]
            }]
        }]
    },
    {
    "id": "class211123",
    "url": "/index.html#/components/11123",
    "name": "Subassembly B1 - 3",
    "color": "#f9ba72",
    "children": [{
        "id": "class211122",
        "url": "/index.html#/components/11122",
        "name": "Subassembly B1 - 2",
        "color": "#ef8962",
        "children": [{
            "id": "class2111112",
            "url": "/index.html#/components/111111",
            "name": "Physical Element B1 - 1.1",
            "color": "#bed682",
            "children": [{
                "id": "class211121",
                "url": "/index.html#/components/11121",
                "name": "Subassembly B1 - 1",
                "color": "#a66c99",
                "children": [{
                    "url": "/index.html#/products/1112",
                    "id": "class21111",
                    "name": "RRUS01 B1",
                    "color": "#F1F2F7",
                    "size": 10
                }]
                }]
            }]
        }]
    },
    {
    "id": "class211133",
    "url": "/index.html#/components/11133",
    "name": "Subassembly B2 - 3",
    "color": "#f5a241",
    "children": [{
        "id": "class211132",
        "url": "/index.html#/components/11132",
        "name": "Subassembly B2 - 2",
        "color": "#e95c38",
        "children": [{
            "id": "class2111112",
            "url": "/index.html#/components/111112",
            "name": "Physical Element B2 - 1.1",
            "color": "#a5c753",
            "children": [{
                "id": "class211131",
                "url": "/index.html#/components/11131",
                "name": "Subassembly B2 - 1",
                "color": "#8f3f7b",
                "children": [{
                    "url": "/index.html#/products/1113",
                    "id": "class21113",
                    "name": "RRUS01 B2",
                    "color": "#F1F2F7",
                    "size": 10
                }]
                }]
            }]
        }]
    }
]},
{
"name": "",
"children": [
    {
    "id": "class11113",
    "url": "/index.html#/components/11113",
    "name": "Subassembly B0 - 3",
    "color": "#fcd2a2",
    "children": [{
        "id": "class11112",
        "url": "/index.html#/components/11112",
        "name": "Subassembly B0 - 2",
        "color": "#f7ba98",
        "children": [{
            "id": "class111111",
            "url": "/index.html#/components/111111",
            "name": "Physical Element C - 1.1",
            "color": "#a5c753",
            "children": [{
                "id": "class11111",
                "url": "/index.html#/components/11111",
                "name": "Subassembly C - 1",
                "color": "#8f3f7b",
                "children": [{
                    "url": "/index.html#/products/1111",
                    "id": "class1111",
                    "name": "RRUS01 B0",
                    "color": "#F1F2F7",
                    "size": 10
                }]
                }]
             }]
```

```
                }]
            },
            {
            "id": "class11123",
            "url": "/index.html#/components/11123",
            "name": "Subassembly B1 - 3",
            "color": "#f9ba72",
            "children": [{
                "id": "class11122",
                "url": "/index.html#/components/11122",
                "name": "Subassembly B1 - 2",
                "color": "#ef8962",
                "children": [{
                    "id": "class111111",
                    "url": "/index.html#/components/111111",
                    "name": "Physical Element C - 1.1",
                    "color": "#a5c753",
                    "children": [{
                        "id": "class11111",
                        "url": "/index.html#/components/11111",
                        "name": "Subassembly C - 1",
                        "color": "#8f3f7b",
                        "children": [{
                            "url": "/index.html#/products/1112",
                            "id": "class1111",
                            "name": "RRUS01 B1",
                            "color": "#F1F2F7",
                            "size": 10
                        }]
                        }]
                    }]
                }]
            },
            {
            "id": "class11133",
            "url": "/index.html#/components/11133",
            "name": "Subassembly B2 - 3",
            "color": "#f5a241",
            "children": [{
                "id": "class11132",
                "url": "/index.html#/components/11132",
                "name": "Subassembly B2 - 2",
                "color": "#e95c38",
                "children": [{
                    "id": "class111111",
                    "url": "/index.html#/components/111111",
                    "name": "Physical Element C - 1.1",
                    "color": "#a5c753",
                    "children": [{
                        "id": "class11111",
                        "url": "/index.html#/components/11111",
                        "name": "Subassembly C - 1",
                        "color": "#8f3f7b",
                        "children": [{
                            "url": "/index.html#/products/1113",
                            "id": "class1113",
                            "name": "RRUS01 B2",
                            "color": "#F1F2F7",
                            "size": 10
                        }]
                        }]
                    }]
                }]
            }
        ]}
    ]
}
```

```
/////////////////////
//  Product menu    //
/////////////////////

[
    {
      "label": "Modular Plan 1",
      "id": 1,
      "children": [
        {
            "label": "Product Type",
            "id": 11,
            "children": [
                {
                    "label": "Product",
                    "id": 111,
                    "children": [
                        {
                            "label": "Product Variant",
                            "id": 1111,
                            "children": [
                                {
                                    "label": "Subassembly",
                                    "id": 11111,
                                    "children": [
                                        {
                                            "label": "Physical Element",
                                            "id": 111111,
                                            "children": [
                                                {
                                                    "label": "Logical Element",
                                                    "id": 1111111
                                                }
                                            ]
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
      ]
    },
    {
      "label": "Modular Plan 2",
      "id": 2,
      "children": [
        {
            "label": "Product Type",
            "id": 21,
            "children": [
                {
                    "label": "Product",
                    "id": 211,
                    "children": [
                        {
                            "label": "Product Variant",
                            "id": 2111,
                            "children": [
                                {
                                    "label": "Subassembly",
                                    "id": 21111,
                                    "children": [
                                        {
                                            "label": "Physical Element",
                                            "id": 211111,
                                            "children": [
                                                {
                                                    "label": "Logical Element",
                                                    "id": 2111111
                                                }
                                            ]
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
      ]
    }
  ]
```

```
//////////////////
// Elements     //
//////////////////

{
    "elements":{
        "labels": [
            {"value":"Name"},
            {"value":"Type", "class": "filter-select filter-exact", "dataplaceholder": "Select"},
            {"value":"Basic Function", "class": "filter-select filter-exact", "dataplaceholder": "Select"},
            {"value":"IBW DL"},
            {"value":"IBW UL"},
            {"value":"Nr of Transmitters"},
            {"value":"Nr of Receivers"},
            {"value":"Power"},
            {"value":"Development Start"},
            {"value":"Release Date"},
            {"value":"Valid To"},
            {"value":"Development Time"},
            {"value":"Geometry"},
            {"value":"Used in Products"},
            {"value":"Used in Elements"}
        ],
        "data":[
            {
                "id": 11111,
                "data": [
                    {"value": "Subassembly B0 - 1"},
                    {"value": "Subassembly"},
                    {"value": "Basic Function X"},
                    {"value": "30"},
                    {"value": "30"},
                    {"value": "2"},
                    {"value": "2"},
                    {"value": "30"},
                    {"value": "2014-01-01"},
                    {"value": "2014-06-01"},
                    {"value": "2016-06-01"},
                    {"value": "1000"},
                    {"value": "X"},
                    {"value": ""},
                    {"value": ""}
                ]
            },
            {
                "id": 111111,
                "data": [
                    {"value": "Physical Element B0 - 1.1"},
                    {"value": "Physical Element"},
                    {"value": "Basic Function X"},
                    {"value": "30"},
                    {"value": "30"},
                    {"value": "2"},
                    {"value": "2"},
                    {"value": "30"},
                    {"value": "2014-01-01"},
                    {"value": "2014-06-01"},
                    {"value": "2016-06-01"},
                    {"value": "1000"},
                    {"value": "X"},
                    {"value": ""},
                    {"value": ""}
                ]
            }
        ]
    }
}
```