INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

USING PLM AND TRADE-OFF CURVES TO SUPPORT SET-BASED CONVERGENCE OF PRODUCT PLATFORMS

Christoffer LEVANDOWSKI, Anders FORSLUND, Hans JOHANNESSON
Chalmers University of Technology, Sweden

ABSTRACT

Platforms may be used as an enabler for offering a variety of products to the market, while keeping the development cost down. Reusing design knowledge is a key concept of platforms, whether concerning reusing parts, ideas, concepts or technologies. In set-based design, trade-off and limit curves are an enabler to store knowledge about technologies developed earlier, and to highlight knowledge-and technology gaps.

This paper describes how trade-off curves derived from technology development may be used to incorporate technology knowledge in a product platform. The product used as an example is a Turbine Rear Structure of a jet engine.

Trade-off curves and a product platform based on the Configurable Component concept is implemented in a PLM architecture, integrating a Product Data Management system, a Computer Aided Design tool, two Computer Aided Engineering tools and a configurator. The analysis combines the trade-off and limit curves with CAE tools to create a comprehensive analysis of the set of possible solutions. The results are presented to the engineer as a means to aid in the convergence process through elimination of bad solutions from the set.

Keywords: product families, product lifecycle management, lean, set-based design, technology platforms

Contact:
Christoffer Levandowski
Chalmers University of Technology
Department of Product and Production Development
Gothenburg
41296
Sweden
levandow@chalmers.se

1 INTRODUCTION

Manufacturing companies strive towards efficiency in production and development (Meyer and Lehnerd, 1997). Reusing previously developed design knowledge in new designs and technologies is a common approach to achieve this efficiency in development. Using a platform as a means for knowledge reuse, where knowledge, components and systems are reused has received a lot of attention the past decade (Jiao, et al., 2007). It is not only a way to gain benefits of scale in production, but also benefits in development (Gonzalez-Zugasti, et al., 1999, Jiao, et al., 2007, Meyer and Lehnerd, 1997, Robertson and Ulrich, 1998).

Trade-off and limit curves are tools used in technology development for storing knowledge about technologies developed earlier, and to highlight knowledge- and technology gaps. In contrast to technology development, parameterized products and product structures common means to model product platforms. When these two lifecycle stages must interact, the difference in how the design knowledge is represented may become a problem. For example, a design set convergence process has to take into account limitations of the technologies used while assessing different design solution. This interaction is particularly problematic when developing platforms, rather than single products, because of the complex interactions a platform imposes on the design.

For long, using IT tools has been a way to manage knowledge and support the business processes, where knowledge is stored as product data (Abramovici, 2002). No software tool is the perfect fit for all business processes of a company. However, the combined capabilities of different tools may very well satisfy the needs. Furthermore, no single business process is able to take on the entire company. Thus, numerous processes and software tools resides side by side. Product information is used and created across the lifecycle of the product, as well as the different disciplines and organizational functions, which often results in a horde of different expert tools. Each of them is an expert in what they are designed to do and are therefore most vital to the business. Consequently, processes, information, tools and people need some way to interact to achieve continuity throughout the product lifecycle. Managing complex platforms requires a lot from the business and Product Lifecycle Management (PLM) has proven to be a useful tool.

This paper discusses how trade-off curves can be used as the link between technology platform and product platform, and how this can be implemented in a PLM environment. The benefits of a platform are many, but depend not only on the theoretical background, but also on the actual implementation. Several different guidelines propose allowing different solutions for different parts of the organization (Rangan, et al., 2005), thus opens up for describing technology platforms and product platforms in different ways using different systems.

Two research questions summarizes the focus for this study:

> RQ1: How can trade-off curves from technology development be integrated into a product platform?
>
> RQ2: How can trade-off curves be implemented in a PLM environment to support the convergence process of new designs?

A case study was performed, in order to give an answer to the posted research questions. The case was set up with data from a aerospace company. Like most aerospace components, the studied product, is characterized by low production volumes using customized solutions for each customer. However, the customization of the product variants is often within the same design solution, i.e. varying design parameters makes the difference between product variants. This makes the product a good candidate for set-based design. The studied company, which has been studied extensively previously, has expressed a need for using both a technology platform, and a product platform. Further, the company has a tradition of using trade-off curves in their development. The PLM architecture that was developed for the case is based on to the industry commonly known IT-systems. The exception is the software used for modeling the product platform.

2 PLATFORMS AND THEIR SUPPORT

A platform is commonly associated with reusing previously designed parts in future products to benefit from the economies of scale. On a more general level, Robertson and Ulrich (1998) define a platform as the “collection of assets that are shared by a set of products”. Using this definition, a platform can mean more than just reusing parts. Meyer and Lehnerd (1997) has a slightly less abstract view of a platform, and states that platform is “a set of subsystems and interfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced", thus bringing it to a level where it is obvious that the platform is dedicated to products.

Meyer and Lehnerd (1997) identify three different levels of a platform: the common building blocks, very similar to what is presented to be included in a technology platform (Jolly and Nasiriyar, 2007); the product platform itself according to their definition; and variants generated from the product platform that together constitute a product family. A technology platforms represent, in a sense, the core competency for technology-based companies, which does not lend itself to the building block modules and interface structure of product platforms (McGrath, 2001).

Generally, a company develops their platform in parallel with product variants and technology development (Wheelwright and Clark, 1992). Even though run in parallel, the general idea is that a technology platform shall, in one of its uses, act as a basis for developing the product platform, the same way the product platform acts as basis for configuring product variants.

The primary result of technology development is usually differs from the result of product development. Typically, technology development has a fuzzy goal of building knowledge or demonstrating feasibility, while product development has a sharp goal of resulting in a commercial product (Nobelius, 2002).

2.1 The relationship between product and technology platform

It is apparent that different stages in the product lifecycle require different support. Yet, there need to be a unified approach on how to leverage from the created knowledge throughout the lifecycle. To achieve efficiency across a lifecycle, several business processes run in parallel (Prasad, 1996). This type of *concurrent engineering* allows for example technology development, product development and manufacturing development to start earlier than in pure sequential development, thus shortening the lead time. On the other hand, it requires integration of teams, tools and product information. Having reusable digital product and process models, such as seen in a platform is a way to facilitate concurrency (Prasad, 1996).

Technology platforms and product platforms are described differently, but to achieve efficiency in product development, the results from technology development need to be easily integrated into the product development projects. On a systems support side, adopting such an approach would ultimately lead to integration challenges, whether they be manual or automatic through interfaces between IT applications. As of today, there is no clear picture of which approach provides the best solution for integrating technology platform and product platform, but several different approaches are possible.

Product Lifecycle Management (PLM) is a business approach that aims to integrate the business processes of an organization, as well as managing the information generated during the lifecycle (Stark, 2005). It is widely recognized as a business approach for fast and efficient product development (Grieves, 2006, Ming, et al., 2005, Stark, 2005). CIMData (CIMdata, 2010) defines it as a business strategy for collaborative creation, management, and use of product definition information, spanning from concept to end of life of a product or plant, and integrating people, processes, business systems, and information. Thus, PLM can be used to tie the different lifecycle stages, and facets thereof together.

Zimmerman (2008) defines a PLM architecture using the Zachman framework, as an IT-centric enterprise architecture, comprising of several different layers, or sub-architectures. The application layer, which is interesting when managing several IT applications with different purposes, assigns which tasks are to be performed by what application (Catic, 2011). However, PLM is not just an IT system. Stark (2005) argues that there are several different parts of PLM, such as the engineering methods and processes, the organization, the product and the product information and IT systems that all need to be considered and coordinated. Svensson et al. (1999) comes to the same conclusion and states four views: processes; information; systems and roles, all of which need to be considered to create a complete PLM architecture.

2.2 Describing product and technology platforms

As a part of set-based design concurrent engineering, trade-off curves are used to document design knowledge for future use (Sobek, et al., 1999). This type of design knowledge then constitutes solid ground on which to build future designs. If managed and maintained, this is one example of a technology platform. The common denominator for trade-off curves is that they model a trade-off between two or more different parameters, typically two that cannot be optimized simultaneously. A subset of the trade-off curve is the limit curve, which marks the limit for what is possible, or safe to do with current technologies. Trade-off curves are used in several lifecycle stages, for example product planning (Burke, 1988) as an aid in deciding what the new product shall be able to do and detailed design for optimizing solutions for multiple criteria. A core element in set-based concurrent engierring is using trade-off curves to rule out infeasible designs, given a set of requirements. While trade-off curves may be produced in technology development, and therefore may reside in a technology platform as technologies, they are used in product development. The trade-off curves may in some cases be connected to the detailed design of a particular system, and therefore reside in close connection to the system description, thus in the product platform.

A cornerstone in describing product platforms is representing variability of the platform. Several approaches to configurable platforms have been suggested in research (Erens, 1996, Mannisto, et al., 2001, van Veen, 1991).

One such approach – the one that will be subject in this paper – is the Configurable Component (CC) concept (Claesson, 2006). A platform described using the CC concept consists of several autonomous systems, each described by a CC object. CC objects can *use* other CC objects to compose themselves. The Configurable Component concept has a great deal in common with a modular product platform, allowing for concurrency while developing the different modules or subsystems (Prasad, 1996), but aims to support a platform approach based on the concept of subsystems and concepts, rather than reusing parts. Each subsystem is configurable to fit a variety of contexts and fulfill the same function in each context. A CC object may represent, for example, an entire car, a front door or a rear view mirror. Essentially, CC objects do not represent merely one type of car door, but rather every door in a product platform – being a model of a system family. The ability to insert different parameters results in a multitude of variants and as a result the door will look or behave differently.

The configurable *design solutions* are tightly connected to the backbone of the CC: the *design rationale* (Figure 2), which is a number of enhanced function-means trees where sets of design solutions are the means. It also consists of the *functional requirements* and *constraints* (Schachinger and Johannesson, 2000). Each functional requirement has a bandwidth within which it can vary, and is solved by one design solution. To answer to the bandwidth of the functional requirement, the design solution in itself has a bandwidth within which it can vary. To cover the entire bandwidth of the functional requirement, it is sometimes necessary to switch between different design concepts. In other words, the functional requirements have a bandwidth (a parameter range), which is met by a set of design solutions (a concept range) which each and every one also has a bandwidth (a parameter range) (Wahl and Johannesson, 2010).

3 A PLM ARCHITECTURE FOR INTEGRATING TECHNOLOGY AND PRODUCT PLATFORMS

The process of narrowing down a set of design solutions should be based on facts rather than assumptions. Within that, eliminating a concept from a set can be done for a number of different reasons for example, if a solution is found completely infeasible in manufacturing. However, solutions that are feasible, but are worse than all other concepts on every performance criteria, can also be eliminated.

This case aims to illustrate how a PLM architecture can use the trade-off curves of technology platform in combination with advanced analysis tools to assess the total performance of a Turbine Rear Structure (TRS) in order to provide a designer with the information he or she needs to eliminate underachieving concepts. To consolidate the analyses and manage the data, a PLM architecture has been set up.

The performance criteria that the TRS is evaluated on are listed below:

- Pressure loss
- Buckling load factor
- Thermal stress
- Over Turning Moment (OTM)
- Shear Compliance
- Geometric Stability

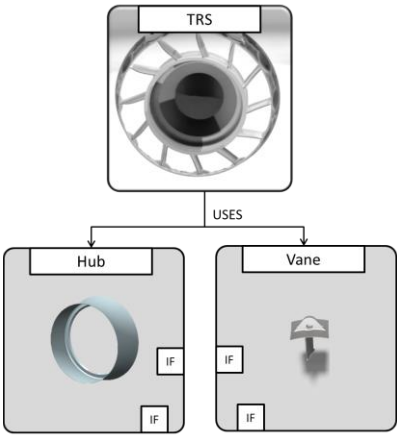

Figure 1.The Turbine Rear Structure is composed by a number of vanes. Depending on the manufacturing concept, the assembly may or may not include a Hub as well.

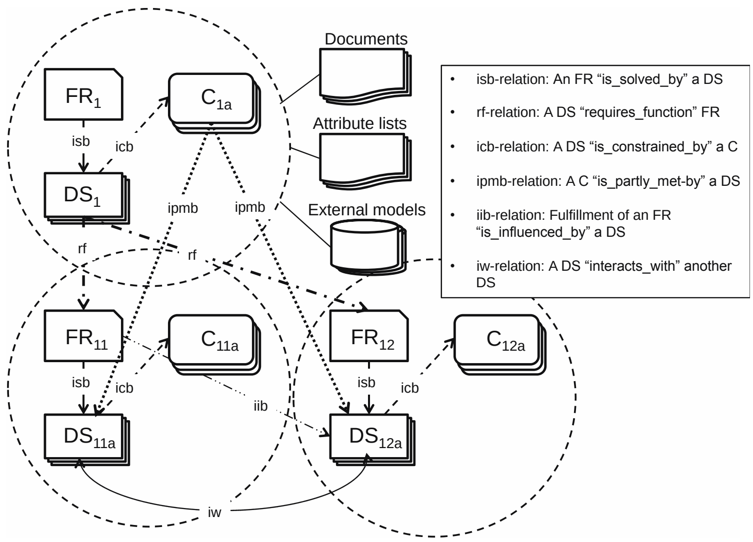

Figure 2. An enhanced function-means tree where a set of design solution together solves the bandwidth of the functional requirements on each level.

A number of different analyses are needed to assess the performance criteria:
(1) Aerodynamics analysis – evaluates the aerodynamic performance of the solution. Specifically, the pressure loss over the TRS and the velocity angle at the outlet are calculated. Further, aero surface temperatures are calculated and fed into the subsequent thermal analysis.
(2) Thermal analysis – calculates the material temperature from given boundary surface temperatures. The results of the thermal analysis are used to calculate thermal stress.
(3) Thermal stress – The recurring thermal loads on the frame create large stresses in the material. This is a limiting factor for product life. Consequently, the thermal stress gives an indication of estimated life. Centerline shift, the movement of the motor shaft centerline because of thermal expansion, is also calculated.
(4) Ultimate stress – assesses whether the turbine structure can withstand extreme events, such as a loss of a fan or turbine blade, or a wheels-up landing. The engine does not need to be operational after such an event, but the engine must not separate from the wing, and no parts should be lost. Ultimate stress is measured on the primary and secondary load paths.
(5) Shear compliance – calculating the inverse of the stiffness of the product, when a unit load is acting on the bearing housing. Compliance is chosen instead of stiffness in order to consistently define the output as something that should be minimized.
(6) Overturning moment – similar to shear compliance, but instead of a force, a torque on the bearing housing around the pitch axis.

3.1 Modeling the product and technology platform

The product (Figure 1) is described using the CC framework, originally developed by Claesson (2006), comprising of several different systems, represented by three different CCs: a TRS complete, representing the product as a whole, an H-section, used to assemble the product using H-sections alone and a T-section and a Hub used to assemble the product with T-sections and an inner ring. Though being different fabrication concepts, the geometry of the final assembly will be the same, thus the same trade-off curves apply.

Each system also has a parameterized CAD model that describes the geometry of that particular system. The geometry, as well as the CC itself has a bandwidth, derived from trade-off curves and other information from a technology platform wiki, which describes the range of what the design solutions can and cannot do. Apart from the geometry description, there is also a design rationale connected to each part. The requirements in this case reside in the TRS complete (Figure 1). The product can vary in two dimensions: the number of guide vanes, and the fabrication concept (T- or H vane). The connection to the technology platform is modeled by connecting CCs to trade-off curves.

For this case, the technology platform consists of two parts, one part realized in a wiki describing technologies and how the can be used etc. The other part is a database with trade-off curves, which have been developed as a part of technology development projects. They are generic in the sense that they cover a wide range of different applications, rather than aiming at a specific product. As new knowledge about the technology emerge, e.g. new material applications, it is implemented in the trade-off curves, given that it has reached appropriate maturity. Trade-off curves are typically generated through physical or virtual testing and using equations describing the product behavior and performance. The curves are modeled using Microsoft Excel for both visual and numerical representation.
Figure 3 illustrates the first trade-off, a trade-off between mass (in discreet steps based on the number of vanes that are used) and stiffness. The stiffness is the reverse of shear compliance, which is one of the performance criteria that are to be evaluated. The second trade-off, illustrated in Figure 4, is between the geometric stability and the mass, again discreetly expressed with the number of vanes. The figure includes two different curves, one for each manufacturing concept.

3.2 The process for configuring product variants

The process of bringing about information of all the performance criteria, in this case a PLM workflow, depend very much on the performance criteria themselves. Preparing this process is something that is done when developing the platform and the IT support for it. The focus is to certify that all performance criteria can be evaluated, in this case six, but in a real case several more. Both trade-off curves are described in a computer interpretable way in a database, along with a description where it is applicable.
Figure 5 illustrates the process required to generate the evaluation information in an IT-context. As means, the different systems are mapped to each activity. As seen in Figure 5, two of the performance criteria are derived using proven knowledge from the technology platform, expressed as trade-off curves. The rest are calculated using a CAE tool.

3.3 System Architecture

Each analysis activity is mapped to a PLM architecture system component that performs the activity. The mapping is basically a realization of the technologies (analysis technologies, and information management technologies) described in the technology platform wiki. The configuration itself is also considered an activity, and is managed by the in house developed configurator tool Configurable Component Manager (CCM). Further, there are several supporting activities to complete the process (Table 1). The connection between technology platform and product platform is found in CCM,

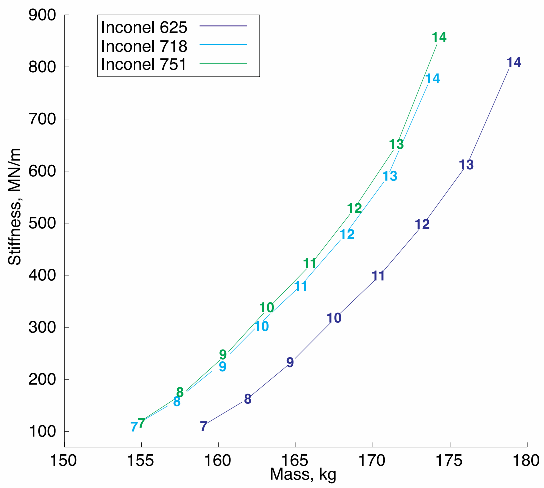

Figure 3: Trade-off curve showing a trade-off between mass and stiffness with current available technologies. The different lines represent different material.

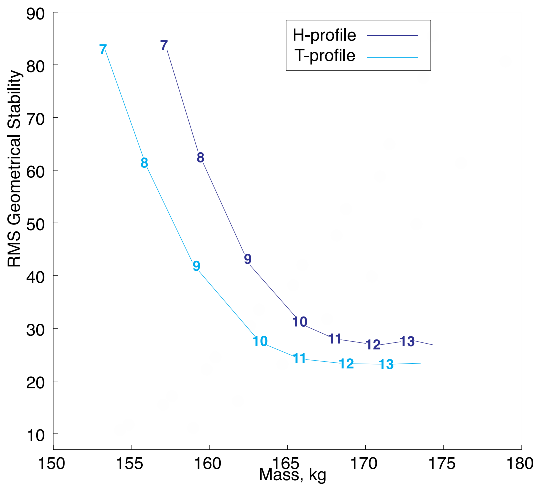

Figure 4. Trade-off curve showing a trade-off between Geometric Stability, Expressed in RMS AND Mass, using current manufacturing technologies. The different lines represent different Manufacturing Concepts.

Microsoft Excel and Share-A-space as mirror images as these are lifecycle-stage transcendent.

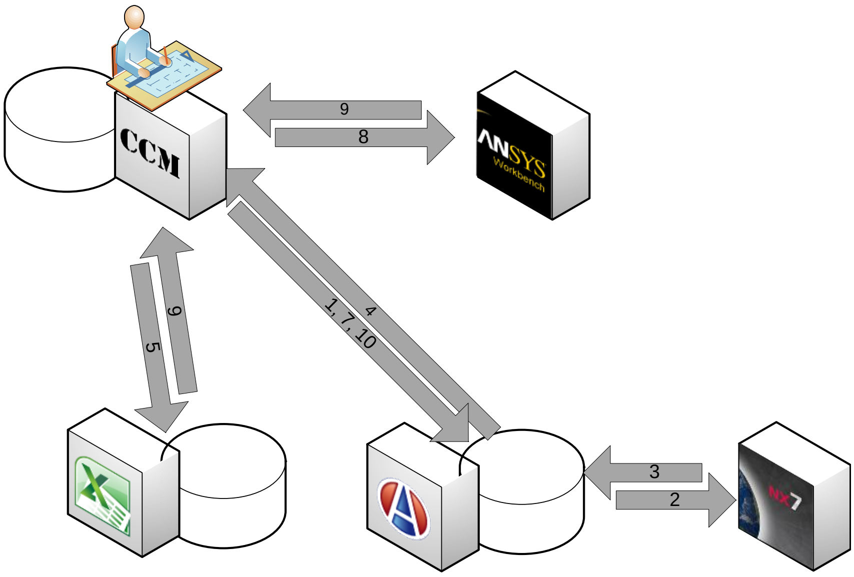

Figure 5. PLM architecture system setup, based on the activities needed to assess the requirements on the frame structure. The interface for the engineer is CCM. The arrows represent the process, and the boxes represent systems used in the process.

Table 1. Mapping between activities and PLM architecture system component.

Activity	PLM architecture system component
Perform Multi Criteria analysis	Ansys Workbench
Manage Configuration	CCM
Store Product Platform information	CCM + Share-A-space
Store Technology Platform information	Excel database
Draw CAD files	Siemens NX7
Store CAD files	Share-A-space

3.4 Executing the platform

As the product platform is prepared as described above, it is now possible to fast be able to generate information on how well product variants in the set meet the new requirements proposed by the customer.

The full system architecture is shown in Figure 5. The analyses and data management is done automatically, and initiated by CCM. The two parameters of the set are varied to create the solution space, resulting in a total of six different configurations. These concepts are then created in the PDM system (1), and the correct CAD files are created, based on the generic one, that describes the whole set (2). The respective CAD files are connected to the concepts in the PDM system (3). The PDM system then sends back the six different concepts' CAD models to CCM (4).

The first two performance criteria are assessed through accessing the (5, 6) trade-off curves, as described in section 3.1, the result is stored in the PDM system connected to each concept (7). The CAE analysis activity is performed by Ansys Workbench, which upon request (8) uses the CAD files of the different concepts to perform a multi-criteria analysis of all concepts. Ansys analyzes pressure loss, buckling load factor, thermal stress, and OTM for the entire solution set (six concepts). As the result is returned (9), it is both stored in the CCM database display it to the designer as well as in the PDM system (10) for reuse at other times.

After all the analyses are done, the result is consolidated in a graph showing the different concepts as lines as shown in Figure 6. This sheet may act as a foundation for discussion and decision in which concept to eliminate and which to pursue. The complete evaluation is finished in a matter of hours, depending on the granularity of the analyses. Further, the more information that is stored as trade-off curves, the quicker the process.

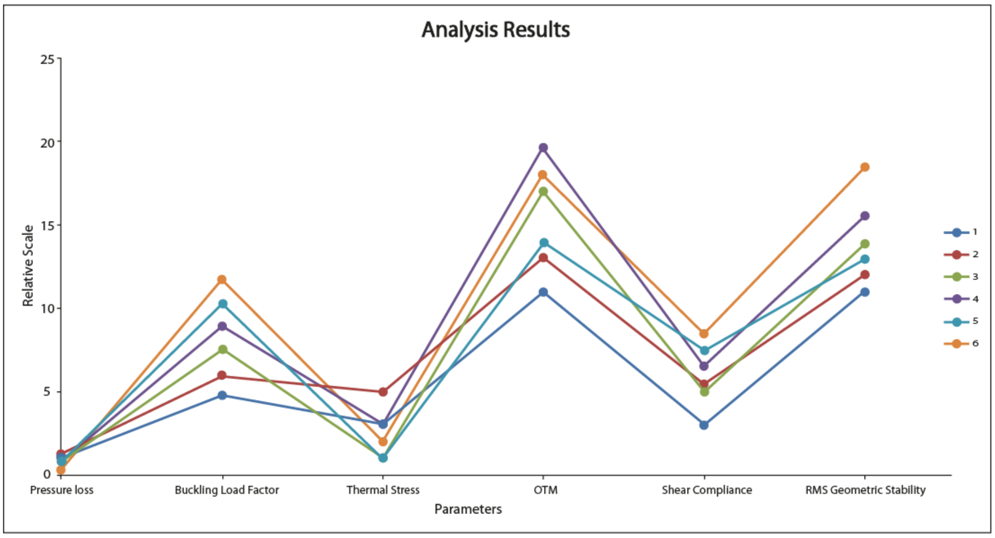

Figure 6. Results from the analysis, in comparable numbers. Each concept is represented by a polygon. OTM: over-turning moment.

4 DISCUSSING BENEFITS AND PREREQUISITES

This chapter discusses the result of the case study in light of the posted research questions. Some major issues are also conferred.

4.1 Shortening the lead time

this case aims to investigate the mere possibility of using trade-off curves as the link between the technology platform and the product platform, implemented in a PLM architecture. Thus, the focus is not to measure the exact amount of saved time in development. However, an essential principle in the design process is that designers with the proper understanding of the design context make better decisions (Hansen and Andreasen, 2002), thus one can argue that automating steps in the gathering of information for the elimination process and leaving the important decisions up to the designer does make the process more efficient. The designer will have time for value adding activates as compared to tedious keyboard mashing and testing of different possible configurations before finding a feasible solution. The information flow is automatic, thus the architecture possesses the ability to call the API of the software to fully automate the process. The design decisions are still up to the designer, who in that case would have CCM as the only GUI, and every other activity would run automatically.

The approach of using the trade-off curves in the technology platform compares well to using just heavy analyses. The time saved by using a technology platform rather than using only CAE analysis depends on the analysis time, and increases with the size of the design space. Also, the time savings in having to CAD everything and manage the different data flows by hand are even greater.

4.2 More work early, less work later

Set-based concurrent engineering and platform-based design both boil down to a front loaded process. The effort needed in the beginning of the process, creating and preparing the platform, is well compensated for with the ability to produce correct information about product variants at a much more rapid pace than before. This gives the advantage of faster and more accurately being able to, for example, answers to customer quotes, and thereby gain leverage toward competitors.

Worth mentioning is that true set-based concurrent engineering is achieved when bad parts of the design space is sliced of, rather than generating point solutions and then eliminating them (Sobek, et al., 1999) as done in this case. The inability to do so is rather due to the limits of the CAE tools than the PLM architecture. Unfortunately, a new era of CAE tools will have to emerge to fully support design spaces, as today’s commercial tools are optimized for point solution. There are some good examples for design space exploration, for example from the optimization community (De Weck, 2004), where modeling the physics of a product is essential for exploring the design space.

4.3 Prerequisites for usefulness

In terms of usefulness of the presented result, it may be used to distinguish low performing concepts. For example, one may conclude that the blue concept seems to outshine the rest of the concepts on almost all points, however, not all of them. Instead, the curves can be used to eliminate the red concept that does not stand out on any of the performance criteria.

The possibility to use this type of tool rests upon two assumptions. First, *the technology platform is already created and expressed as computer interpretable trade-off curves.* This assumption, given that a technology platform exists, is not far-fetched. It does however require the technology development process to include trade-off curves as a deliverable, and that they are stored and maintained in a structured way. Also, exhaustive enumeration of product configurations is only possible for the simplest systems, thus the approach may have to be proved for larger design spaces.

Second, *the product platform is already created and expressed as a collection of Configurable Components and parameterized CAD geometry*. However unlikely it may be that a company would have implemented the CC-concept to its full extent, modern CAD geometries are often parameterized and capable of communicating with external software. The effort of creating a product structure based on the CC-concept is then fairly effortless. There are other strategic decisions that need to be made, such as adopting set-based concurrent engineering, and working with technology platforms.

Further, how the information is displayed can very much influence the actual usefulness of the results. The proposed way has not been verified, and there is extensive research on how to display information. In reference to the research, within that field there are most likely better ways to display the information that eases the convergence process even more. However, the information is there, the rest is a matter of format.

5 CONCLUSIONS

The proposed IT architecture does show that it is possible to create an IT support that aids the designer in the process of converging designs towards a feasible solution. Through combining trade-off curves and CAE tools, in comparison to just using extensive analyses, the lead-time for analysis can be substantially reduced. However, though IT-systems supposedly do a lot of the work here, the use of platforms with trade-off curves will require a whole new way of working. Besides the implementation of new IT systems, using platforms requires the processes to be front loaded, preparing platforms in order to later be able to harvest the fruits of them.

The first research question, *Can trade-off curves be used as the link between technology platform and product platform?*, can in this case be seen as answered. It is possible, but comes with a number of prerequisites, such as how the platforms are modeled. Further, this is a simple case varying only two design parameters and using fairly simple trade-offs. However, it is safe to say that a computer can handle much more complex and greater numbers of trade-offs with ease.

The question *How can a trade-off curves be implemented in a PLM environment to support the convergence process of new designs?* is given an answer through the proposed PLM architecture. The platform modeler together with the technology platform, CAE tool and PDM system provides a fair display of the performance of several solutions within the set, enough to narrow down the set. The visualization of the information is still to be researched.

REFERENCES

Abramovici, M. (2002) 'Status and Development Trends of Product Lifecycle Management Systems', *International Conference on Integrated Product and Process Development*, November 21-22, Wroclaw, Poland.

Burke, W. J. (1988) 'Trade off methods in system planning', IEEE Transactions on Power Systems, vol. 3, no. 3, pp. 1284-1290.

Catic, A. (2011) 'Knowledge-based Engineering in Product Development Processes - Process, IT and Knowledge Management perspectives', Ph.D. thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg.

CIMdata (2010) 'Product Lifecycle Management (PLM) Definition', Retrieved March 2nd, 2011, from http://www.cimdata.com/plm/definition.html.

Claesson, A. (2006) 'A Configurable Component Framework Supporting Platform-Based Product Development', Ph.D. Thesis, Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

De Weck, O. L. (Year) 'Multiobjective optimization: History and promise', *Proceedings of Invited Keynote Paper, GL2-2, The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan*.
Erens, F. J. (1996) 'The Synthesis of Variety: Developing Product Families', PhD Thesis, Enidhoven University of Technology, Eindhoven, The Netherlands.
Gonzalez-Zugasti, J., Finch, W., Otto, K. and , p.-. (1999) 'Set-based Product Platform Requirement Definition', *12th International Conference on Engineering Design – ICED 99*, August 24-26, Munich, Germany.
Grieves, M. (2006) *'Product Lifecycle Management: Driving the Next Generation of Lean Thinking'*, New York, New York, USA: McGraw-Hill.
Hansen, C. T. and Andreasen, M. M. (2002) 'Two Approaches to Synthesis Based on the Domain Theory', in (ed.) *Chapter in Engineering Design Synthesis,* London: Springer-Verlag.
Jiao, J., Simpson, T. W. and Siddique, Z. (2007) 'Product Family Design and Blatform-Based Product Development: a State-of-the-Art Review', 18, Journal of Intelligent Manufacturing,
Jolly, D. R. and Nasiriyar, M. (2007) 'Technology Platform Exploitation: Definition and Research Boundaries', *16th International Conference on Management of Technology - IAMOT 2007*, May 13-17, Miami, Florida, USA.
Mannisto, T., Peltonen, H., Soininen, T. and Sulonen, R. (2001) 'Multiple Abstraction Levels in Modelling Product Structures', Data & Knowledge Engineering, vol. 36, no. 1, pp. 55-78.
McGrath, M. E. (2001) *'Product Strategy for High-Technology Companies: Accelerating Your Business to Web Speed'*, New York ; London: McGraw-Hill.
Meyer, M. H. and Lehnerd, A. P. (1997) *'The Power of Product Platforms: Building Value and Cost Leadership'*, New York, New York, USA: Free Press.
Ming, X. G., Yan, J. Q., Lu, W. F. and Ma, D. Z. (2005) 'Technology Solutions for Collaborative Product Lifecycle Management - Status Review and Future Trend', *The Journal of Concurrent Engineering: Research and Applications*, Dec,
Nobelius, D. (2002) 'Managing R&D Processes - Focusing on Technology Development, Product Development, and their Interplay', Ph.D. Thesis, Operations Management and Work Organization, Chalmers University of Technology, Gothenburg, Sweden.
Prasad, B. (1996) *'Concurrent Engineering Fundamentals'*, Upper Saddle River, N.J.: Prentice Hall PTR.
Rangan, R. M., Rohde, S. M., Peak, R., Chadha, B. and Bliznakov, P. (2005) *'Streamlining Product Lifecycle Processes: a Survey of Product Lifecycle Management Implementations, Directions, and Challenges'*:
Robertson, D. and Ulrich, K. (1998) 'Planning for Product Platforms', Sloan Management Review, vol. 39, no. 4, pp. 19-31.
Schachinger, P. and Johannesson, H. (2000) 'Computer Modelling of Design Specifications', Journal of Engineering Design, vol. 11, no. 4, pp.
Sobek, D. K., Ward, A. C. and Liker, J. K. (1999) *'Toyota's Principles of Set-based Concurrent Engineering'*:
Stark, J. (2005) *'Product Lifecycle Management: 21st Century Paradigm for Product Realisation'*, London, UK: Springer.
Svensson, D., Malmström, J., Pikosz, P. and Malmqvist, J. (1999) 'A Framework for Modelling and Analysis of Engineering Information Systems', *ASME Design Engineering Technical Conferences*, September 12-15, Las Vegas, Nevada, USA.
van Veen, E. A. (1991) 'Modeling Product Structures by Generic Bills-Of-Material', Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.
Wheelwright, S. C. and Clark, K. B. (1992) *'Revolutionizing Product Development : Quantum Leaps in Speed, Efficiency, and Quality'*: Free Press: New York, NY, USA.
Zimmerman, T. (2008) 'Implementing PLM Across Organisations', PhD thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg.