NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

Implementing the principles of Set-based Concurrent Engineering in Configurable Component Platforms

Dag Raudberget
School of Engineering, Jönköping University, Sweden
dag.raudberget@jth.hj.se

Peter Edholm
Department of Product and Production Development, Chalmers University of Technology, Sweden
edholmp@chalmers.se

Magnus Andersson
PE Geometry Development, Göteborg, Sweden
magnus.andersson@geomdev.se

Abstract

This paper describes a new design approach that implements the three principles of Set-based Concurrent Engineering by using the concept of Configurable Component modelling. Several case studies has proven the efficiency of Configurable Component modelling as well as the Set-based philosophy, and by combining these two research areas, a computer based modelling of Configurable Component objects is used to support the Set-based philosophy. The approach is demonstrated by a case study that indicates a promising future of combining Set-based Concurrent Engineering with Configurable Component modelling for re-design problems.

Keywords: Configurable Components, Set-based Concurrent Engineering, Design Process, Computer Support.

Introduction

Set-based Concurrent Engineering (SBCE) is a philosophy that enables multiple concurrent concepts during product development [1]. Literature describes it as highly efficient but hard to employ in traditional industry [2], and there are few industrial examples of its application. Configurable Components (CC) is an object oriented way to describe engineering data and specifications for a product platform [3].

Modelling product platforms is a way of describing the functions and characteristics of a product family. Set-based Concurrent Engineering is a way of structuring the design process. In short, CC describes what the product does and SBCE describes how to develop the product. At a glance, it may seem that CC and SBCE have few things in common, but there are no inherent contradictions that hinder SBCE and CC to be used together. Also, the CC concept has features that could support the principles of SBCE.

This paper describes an implementation of the three principles of SBCE [4] by using Configurable Component modelling. It is based on a literature review and a case study of a re-design problem from the aerospace industry. The objective was to investigate if the Set-Based Concurrent Engineering principles could be supported successfully by using the computer support available in the Configurable Component modelling approach.

The research process started with a survey of academic- and other literature in order to find the latest information and examples of successful applications. The conclusion was that the idea of combining SBCE and the CC concept was not earlier described. After that, a framework for using CC to support SBCE was developed. Finally, the framework was demonstrated by the researchers by using data from an on-going industrial case.

Related research

This work builds upon two different fields of research: Set-based Concurrent Engineering that could be seen as a part of product development management, and Configurable Component modelling that can be seen as a part of computer support for product development. Though the fields are disparate, the authors argue that CC concept can support the principles of SBCE.

Set-based Concurrent Engineering

The concept of SBCE is characterized by developing multiple solutions to design problems in parallel. It considers sets of design alternatives rather than a specific design. A "set" denotes a palette of different solutions to a specific function or problem and can be seen as a family of possible designs. As the design evolves, the sets of solutions are gradually narrowed down based on relevant information from customers, manufacturing departments, tests, and other sources. In end, only one solution is left.

The term "Set-Based" is opposed to the term "Point-based" [1], describing the traditional development methodology. In this context, a Point-based design is characterized by an early selection and approval of one "best" specific design, a single point in the solution space. This initial design is then refined, re-worked and sequentially modified until an acceptable solution is found.

SBCE is not a prescriptive methodology. Instead, it relies on three principles [4]:

(1) Map the design space
(2) Integrate by intersection
(3) Establish feasibility before commitment.

The first principle implies to develop sets of design alternatives from the perspective of each discipline or department. Specific designs are not considered alone; instead the disciplines shares all designs with the other departments [5].

The second principle focuses on reducing the sets. This is balanced by the desire to keep the design space unrestricted until sufficient information is acquired to enable design commitments. The intersections between the sets are identified by two different approaches. The first approach is to find the region where members of the individual sets are compatible with each other. The second approach is to find the border lines where the current constraints are satisfied.

The third principle focuses on narrowing the sets gradually while the level of detail of the remaining solutions is increased. The solutions that cannot satisfy the constraints are eliminated, and the weak individual solutions are sorted out.

These principles may seem simple, but has proven themselves valuable in earlier industrial case studies [6], [7].

The Configurable Component modelling concept

Product and production platforms are widely used by companies to, among other things, get economy of scale and to reduce time to market. One problem is that many platform definitions reduce the flexibilities for the engineers and therefore limits the bandwidth of the final products. Several researchers have defined alternative platform descriptions to get more variability in the platform. One such more abstract configurable platform model, proposed by [8], consists of generic bills-of-materials (BOMs) that provide possibilities to describe large varieties of product types and structures. The idea is to define products as “sets of product types“, instead of defining “individual product types“. This concept has been applied and explored further by [9] when developing product families and synthesizing product variants. Männistö et. al. also propose a strategy related to the idea of a generic BOM [10]. They describe a “master BOM”, which is a generic description for many product variants that can be defined and manufactured on the basis of the platform.

The ‘CC platform’, used in this work, is based on autonomous knowledge-carrying configurable generic subsystems, so-called Configurable Components (CCs) [3]. This approach provides much more configuration flexibility than a part-based defined platform. Such configurable and knowledge-based platform architecture is also more robust, as reuse of configurable subsystems instead of reuse of parts makes it possible to have the complete product knowledge contents available for redesign in order to meet new demands.

When adopting the CC approach, a product or a product platform is described by a linked set of configurable components; see Figure 1. Each CC has relations to other CCs and is instantiated by setting values on its variant parameters (VPs) in the ‘control interface (CI)’. Composition of variants is defined by the ‘composition set (CS)’ with its ‘Composition Elements (CEs)’, which use variant parameters, design concept definitions, and design rules to compose an instantiated variant. The result of a composition is, if necessary, transferred to other CCs used or to external applications, like PDM and CAD systems, to fully generate variant-defining information needed to prepare for production. This kind of functionality can be achieved if each CC has the following necessary knowledge about itself:

1. Knowledge about its origin, i.e. what it should do and be, how this is realized, and why the chosen solution is what it is.
2. Knowledge about its interactions with the external environment.
3. Knowledge about how to compose variants of its design solutions by means of its internal resources or by using other external CCs.

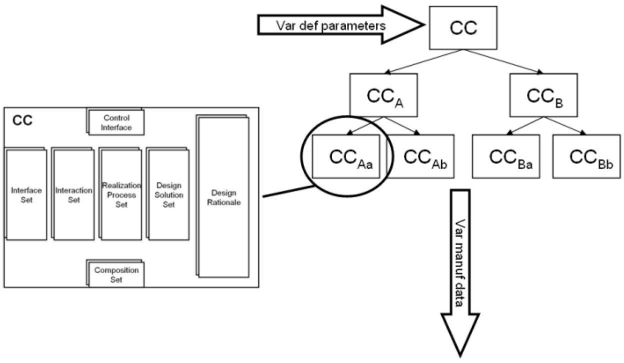


Figure 1. Example of a CC platform.

A framework to implement the principles of Set-based concurrent engineering through Configurable Product Platforms

The suggested framework follows the three principles of SBCE [4].

Implementing principle 1- Map the design space

The first principle of SBCE is Map the design space. This is realized by defining the solution space and the constraints of the product. For the CC objects these constraints are defined in two ways:

i. In the extended function mean tree, where the constraints are used to select the available design solutions to satisfy a specific requirement.
ii. In the overall requirement setup which is used as an input criteria when starting the configuration.

When the constraints are set, all possible solutions could be generated. In this study we use the demonstrator software “CCM”, Configurable Component Modeller to model the product family with CC objects and to generate the different conceptual variant solutions.
The outcome is in this case a number of discrete design solutions/product variants that all *satisfy the overall requirements and constraints*. In this phase the variants are still unconfigured implying that the requirement space is not fixed and that all design parameters are not yet set.

Implementing principle 2- Integrate by intersection

The CC objects include two elements that ensure compatibility between the parts in the set of product variants that satifies the overall requirements; the interface design solution and the interaction design solution. The interaction element strives to establish interfaces between parts that are fully compatible. If no compatible solution is found for a product variant, this will be excluded in the set of possible solutions. This could happen in every step in the product development process as soon as new data is available.

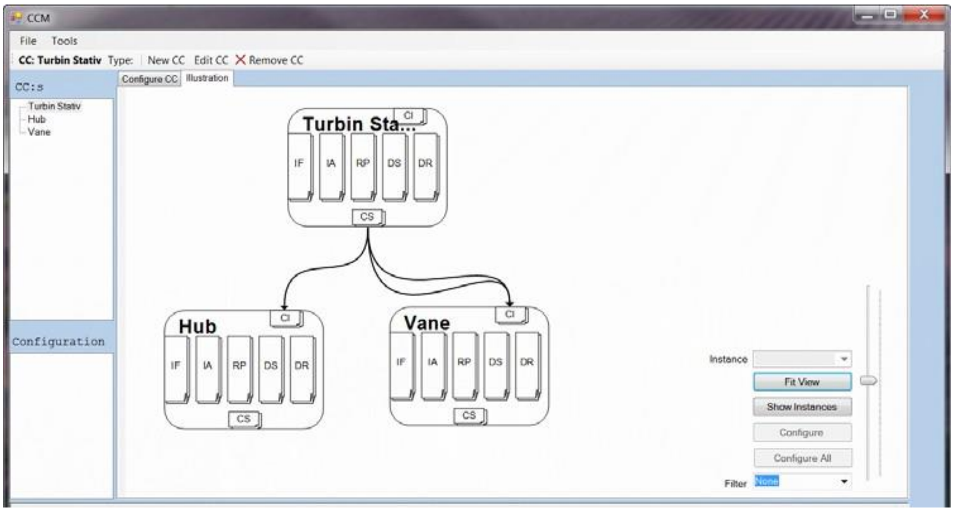
Implementing principle 3- Establish feasibility before commitment

During the subsequent development, more information on the different design alternatives becomes avaliable as the solutions gets more detailed. The configuration routine in CCM communicates with Computer Aided Engienering (CAE) tools to evaluate the performance of the set of product variants. Solutions are then gradually eliminated either due to unfulfilled requirements or by prioritizing certain properties.

Case study

To demonstrate the new framework, data and tools from an on-going industrial research project was used, aimed at improving the efficiency of the design process by introducing tools for CC modelling. Here, the Configurable Component Modeller (CCM) was developed to support research on the CC concept [11].

In the present case study, the configuration was prepared by CCM to support the three principles of SBCE. In CCM, interfaces and interactions between components are modelled and used during configuration which is a different strategy compared to the existing commercial configuration software. Another difference is that CCM delivers a *set of solutions* without choosing or ranking, while many existing configurators aim to find the first variant that satisfies the constraints. The starting point in CCM is a product platform description using CC objects. In this case there are three different CC´s to fully describe a product platform for turbine wheels; Turbine, Hub and Vane (see figure 1). These three CC‘s hold information about all possible solution alternatives that could be configured and what numerical ranges that are allowed on each parameter, the solution bandwidth. This defines the available solution space for the defined platform.



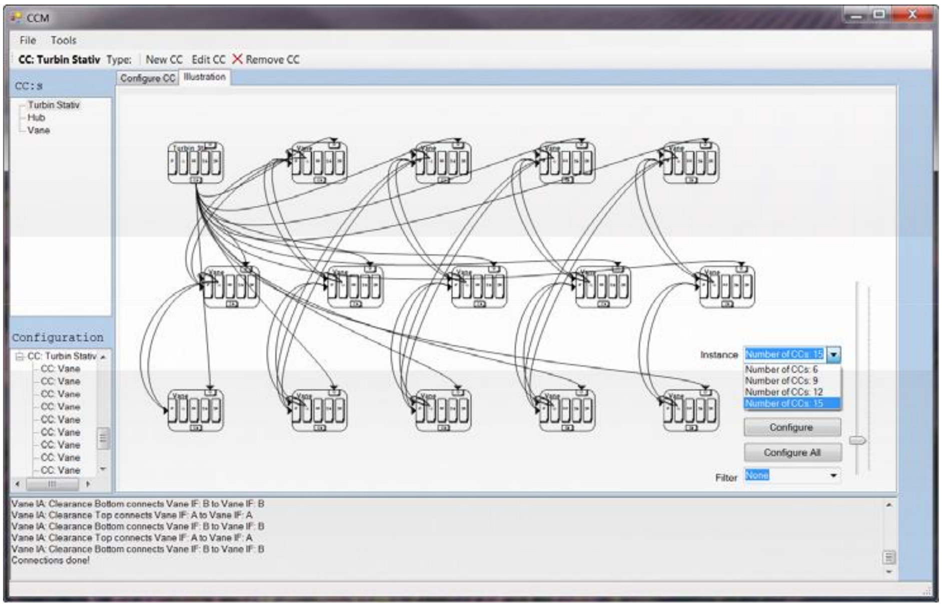
Filter: None

Figure 1. CC platform described in CCM.

Map the design space

When a new development project starts, the overall requirements are set. In this case study they are defined by the customer in form of restrictions specifying product features such as minimum performance values, geometrical shapes, maximum sizes etc.

CCM now performs the first step of configuration, called instantiation, which is creating the design space available that could satisfy the current constraints and restrictions. For this case study CCM generated four different product solution architectures. The result is shown in figure 2, where CCM displays a schematic view of one of the four product solutions: from the platform with three CCS’s, CCM has now generated a product solution with in total 15 objects to satisfy the overall customer requirements.



Instance: Number of CCs: 15

Filter: None

```
Vane IA: Clearance Bottom connects Vane IF: B to Vane IF: B
Vane IA: Clearance Top connects Vane IF: A to Vane IF: A
Vane IA: Clearance Bottom connects Vane IF: B to Vane IF: B
Vane IA: Clearance Top connects Vane IF: A to Vane IF: A
Vane IA: Clearance Bottom connects Vane IF: B to Vane IF: B
Connections done!
```

Figure 2. One conceptual product solution in CCM.

Note that in this stage the product solutions are still conceptual solutions, the detailed parameters are still not set and it is not possible to evaluate the performance of each solution.

Integrate by intersection

The next SBCE principle is Integrate by intersection, and in CCM this principle is realized by linking the CC's to the interaction objects, ensuring compatibility between all interfaces. The "Interface" and "Interaction" functions within the CC concept tests that among the set of possible product variants, only the valid instances will be completed through the instantiation phase. Incompatible parameter values are removed from the set of active solutions using the configuration rules in CCM.

Establish feasibility before commitment

In the next step of configuration, CCM communicates with various commercially available CAE- tools, such as calculation tools and CAD. The purpose is to gain more information on the performance of the remaining solutions. Here, analysis of the performance of the remaining product solutions is generated and the parameters and configurations that could not satisfy the overall customer requirements are eliminated. In the case study, this resulted in disqualification of one solution and three remaining product solutions. In this stage most of the parameters are set and there exist three almost fully defined product solutions.

Now, the main configuration and elimination work in CCM is done and the performance values simulated in the previous step of the remaining product solutions can be visualized see figure 3 for an example. From this, the project management can trade-off different product properties and further constrain the requirements to step-vise decrease the number of remaining configurations until only one product solution remains to take into production.

	Pressure loss	Buckling load factor	Thermo stress	OTM	Shear compliance	RMS Geometric Stability
◆ 1	1	4.8	3	11	3	11
■ 2	1.3	6	5	13	5.4	13
▲ 3	0.8	7.6	1	17	5	13.9

Figure 3. Example of performance value visualization.

Results

The result of the work is an framework combining SBCE and computer support in the form of CC modelling (see table 1). In this work, the development of a mature product can be seen as an extended configuration process, where base elements are configured and adapted to an existing product structure. Many of the product development steps are simplified in the case study, but still the main principles of SBCE and the main concept of CC is clearly used throughout a product development.

In the new approach :

- CCM is used to generate a number of concepts to a specific product variant that should be configured from the CC platform - CCM generates multiple discrete design solutions/product variants for all existing components. The configurations are based on the valid design parameters for each component. By this the first principle of SBCE, “Map the design space” is realized.
- By using the design solutions “Interface” and “Interaction” within the CC concept, a set of possible product variants are defined through CCM´s instantiation phase. Only the valid instances will be completed and the second principle, “Integrate by intersection” is realized.
- The detailed configuration step in CCM is performed on the remaining solutions by the means of parameter variation, analysis of performance etc. and finally one product variant will emerge, realizing the third and final principle, “Establish feasibility before commitment”.

Table 1. A summary of how the CC platform supports the principles of SBCE.

Principle	**Explanation**	**Embodiment**
Map the design space	Define feasible regions, explore trade-offs by designing multiple alternatives and communicate sets of possibilities.	CCM is used to generate a number of discrete design solutions/product variants to all components and configurations based on the design parameters.
Integrate by intersection	Look for intersections of feasible sets, impose minimum constraint and seek conceptual robustness.	The compatibility between all components is investigated and incompatible parameter values are removed from the set of active solutions using the configuration rules in CCM.
Establish feasibility before commitment	Narrow sets gradually while increasing detail, stay within sets once committed and control by managing uncertainty at process gates.	The parameter range at system level (from a customer perspective) is narrowed step-vise to decrease the number of remaining configurations using constraints and performance analysis in the CC concept.

Conclusions

The objective was to investigate if the principles of Set-Based Concurrent Engineering could be supported by using Configurable Component modelling. This case study shows that it is possible to successfully combine these concepts for re-design problems. When the development of a mature product can be seen as a configuration process, the CC concept in itself can be viewed as a mean to realize the principles of SBCE.

The research is, however, restricted in complexity, but the study creates a basis for further investigations within this area. To evaluate its applicability, more advanced cases needs to be explored. Also, it remains to investigate if the approach can support changes in requirements, product structure and components.

References

[1] Morgan, J.M., and Liker, J.K., "*The Toyota product development system: integrating people, process, and technology*", Productivity Press, New York, NY, 2006

[2] Kennedy, M., Harmon, K. and Minnock, E. "*Ready, Set, Dominate: Implement Toyota's Set-Based Learning for Developing Products and Nobody Can Catch You*", Oaklea Press, Richmond, Va. 2008

[3] Claesson, A.,"*A Configurable Component Framework Supporting Platform-Based Product Development*", PhD Thesis, Chalmers University of Technology, Göteborg. 2006

[4] Sobek, D.K., Ward, A. and Liker, J., "Toyota's Principles of Set-Based Concurrent Engineering", *Sloan Management Review,* vol. 40 no 2, pp67-83, 1999

[5] Ward, A., Liker, J.K., Cristiano, J.J. and Sobek, D.K. ," The second Toyota paradox: How delaying decisions can make better cars faster", *Sloan Management Review*, vol. 36 no 1, pp43-61, 1995

[6] Raudberget, D.,"Practical applications of Set- based Concurrent Engineering in industry", *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 56, no.11, pp. 685-695, 2010

[7] Raudberget, D., "Enabling Set-based Concurrent Engineering in traditional product development", *International Conference on engineering Design*, ICED11, 2011

[8] Van Veen, E.A., 1990, *"Modelling Product Structures by Generic Bills-Of-Material"*, PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

[9] Erens, F.J., 2001, *"The synthesis of variety: developing product families"*, PhD thesis, Eindhoven University of Technology, Eindhoven, the Netherlands, 1996

[10] Männistö, T., Peltonen, H., Soininen, T. and Sulonen, R., 2001, "Multiple abstraction levels in modelling product structures", *Data and Knowledge Engineering*, Vol **36**(1), pp 55–78

[11] Edholm, P., Wahl, A.L., Johannesson, H., and Söderberg, R., "Knowledge based configuration of integrated product and process platforms", *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* IDETC/CIE 2009, San Diego, California, USA, 2009