A design automation development process for building and bridge design

Marcus Sandberg, marcus.sandberg@ltu.se
Luleå University of Technology, Sweden

Robert Gerth, robert.gerth@designevolution.se
Design Evolution Scandinavia AB, Sweden

Emma Viklund, emma.viklund@ltu.se
Luleå University of Technology, Sweden

Abstract

The increasing industrialization and standardization of construction opens up for the field of design automation. Such applications, for buildings and infrastructural products, are starting to appear by using software from the manufacturing industry. A challenge is, however, to develop such design automation applications since approaches combining product analyses, requirements management, and development of product platforms and configurators are lacking for construction. Using a bottom up approach to study existing best practice and create more general approaches is one way. In this paper we study the design automation development process from three cases; 1) edge beams, focusing on requirements management, 2) outer non-bearing walls, considering configurator development, and 3) end frame bridge superstructures, featuring the development of generic analysis procedures. These three processes are analyzed and merged into a more generic process which can be used to guide future developments of design automation applications within construction.

Keywords: Configuration, modularization, design automation, knowledge-based engineering

1 Introduction

Some of the most important decisions are taken during construction design as these decisions directly impact constructability. These decisions depend extensively on designers' experience, that is seldom transferred to other engineers. The main reason is short-term interactions between loosely-coupled job partners, a relationship which does not give much incentive to develop practices, methods, and designs that could be reused between projects (Gadde and Dubois 2010). The use of design automation applications can help designers to reuse successful solutions from earlier projects instead of reinventing the wheel in each project. Thanks to automation it becomes easier to generate several solutions and trying different what-if-conditions than when done manually. But, it is hard to know how to develop design automation applications since comprehensive approaches combining requirements management, modularization and configuration are missing for construction. Methodologies for each part have been presented e.g.: Elgh (2008) on requirements in design automation, Erixon (1998) on product platforms development and modularization, and Jensen (2014) on configuration; although few cases are from the construction industry. Therefore, a methodology should be hands-on and describe how to choose tasks to automate, how to capture and formalize knowledge and how to transform the knowledge into logic and rules for implementation in a software that enables semi-automation of blueprints and documents, specific for the construction project. Further, it is hard for companies to know what tool to develop and how to develop it, especially for companies with little or no experience in design automation. Additionally, for researchers it is hard to reproduce research and build on ideas due to lack of details in the papers regarding how the design automation application was created. Hvam et al. (2014) also point out that there is a need for more empirical based investigations to increase the understanding of the process. In this paper three different approaches are presented and combined into one generic process.

2 Background on design automation

2.1 Modularization and configuration

Product configuration can be described as a simplified design process of a product, or the specification process of a product (Jensen 2014). Throughout the configuration process a particular customer requirement is specified, on the basis of a given framework (platform), and detailed products can be generated. The specification work is based on a generic product platform structure consisting of configurable modules or parts. Ulrich (1995) elaborates on the concept, describing product platform as the arrangement of functional requirements and, their relation to physical components, and the specification of interfaces among the interacting components. By specifying the form of these modules and combining them in different ways the targeted customer segment's needs and values can be met (Jørgensen 2001).

In most cases, the product platform consists of too much information to manage manually, why product configurators are needed to support the specification work. Configurators are software systems that supports the configuration process. Blecker and Abdelkafi (2006) define configurators as “software with logic capabilities to create, maintain, and use electronic product models that allow definition of all possible product options and variation combinations, with a minimum of data entries”. Configurators can be seen as design automation applications specialized toward semi-automatic configuration of modular products. Hence, product configurators manage all necessary information of the product architecture, e.g. information about the modules, their constraints, variation rules and possible combinations, in order to specify a tailored product (Helo 2006). Therefore, the development of the configurator is closely related to the development of the product platform. But, the modularization work must proceed the configurator development.

When a company shall develop a configurator and a product platform, the company must identify the market needs and customer values. Needs and values are then converted into functional and technical specifications for the generic product. The generic product platform structure is decomposed into components and modules, as well as rules for how they should be used to generate customer value (Jensen 2014). This process is often denoted modularization (Erixon 1998). Erixon (1998) presented a methodology called Modular Function Deployment (MFD) for development of modular product platform consisting of five steps: 1) Clarify, customer requirements, 2) Select technical solutions, 3) Generate concepts, 4) Evaluate concepts, and 5) Improve each module. Holmqvist and Persson (2003), divide the modularization process into three steps; (1) decomposition of the product into functional or structural parts; (2) integration of modules and parts into a generic product platform and (3) evaluation of the resulting product’s modular characteristics.

2.2 Requirements management in design automation

Blecker et al. (2004) focuses on how to better ensure a correct understanding of customer requirements by using a front-end software system to guide customers through a product configuration process. Stokes (2001) focus is on capturing and formalizing engineering knowledge and the formalizing part results in a Product Model storing the ‘What’ of a design and a Design Process Model storing the design rationale (the ‘Why’) and ‘How’ of a design automation application. Although the Design Process Model features requirements management it does not detail how similar requirements are found. Though Haug et al. (2012) provides a valuable description of the pros and cons of different development strategies, they do not focus on practical approaches to managing requirements. Documentation of information is discussed by Haug & Hvam (2007) where product variant masters, class diagrams and CRC cards are used. Elgh’s (2008) view on requirements management in a design automation context includes the identification, formulation, allocation, verification, and management of changes of requirements. He uses checklists, classification systems and documentation of the built in logics of the product to ease requirement management and ensure traceability.

2.3 Knowledge-based engineering

The term knowledge-based engineering (KBE), has become a label for automating routine design work within the manufacturing industry. It is named ‘knowledge-based’ because knowledge from engineers is captured, formalized and implemented into a computer-based design automation application. Typically, such design automation applications feature both fully automated tasks as well as semi-automated tasks that require user-interaction and often feature computer aided design and computer aided engineering systems. Stokes defines KBE as “the use of advanced software techniques to capture and reuse product and process knowledge in an integrated way” (2001). According to Lovett et al. (2000) KBE applications feature engineering knowledge, geometry and configuration while La Rocca (2012) see KBE as being between CAD and artificial intelligence. Examples of KBE-like design automation applications have been reported within construction e.g. (Sandberg et al 2008). Stokes has presented the most comprehensive methodology for developing KBE-applications, called MOKA and containing six steps: IDENTIFY determines objectives, scope and a concept level technical specification for the design automation application. JUSTIFY examines commercial, cultural and technical risks. CAPTURE collects the raw knowledge and structures it into the Informal Model. FORMALIZE translates the Informal Model into the Formal Model. PACKAGE involves translating the MOKA Formal Model into code for a KBE application. ACTIVATE involves distribution, installation and use.

3 Cases

This section presents the three processes for design automation development from the same construction company used to develop configurators for edge beams, outer non-bearing walls and end frame bridge superstructures. The case company (Design Evolution) is a subsidiary to one of the major construction-engineering consultancy firms in Sweden. The parent company, Tyréns AB, has 1300 employees and delivers design solutions for all product types and contractors through conventional construction engineering work. The case company develops product platforms as projects and uses a project management method inspired by methodologies such as the 5-step model see (Jensen et al 2014) and Agile software development. The development involves engineering of the product architecture, optimizing product flexibility, product modelling, programming the configurator, and performing tests.

3.1 The development of the edge beam configurator

Viklund et al. (2016) derived a requirement management perspective to modularization based on the work of Erixon (1998) and Elgh (2008). This theoretical base was used as support when describing the case company’s approach to developing product platforms and configurators. Edge beams were used as an example for describing the development process. The process consisted of six steps; identifying requirements, formulating requirements, allocating requirements to technical solutions, generating concepts, verifying, and managing changes (see figure 1). During the case it was shown to be difficult to completely separate the process of developing the configurator from the process of developing the product platform due to their close interrelation. It was also noticeable at the first four process steps were highly interrelated. Iterations (shown with circular arrows in Figure 1) were frequent enough to question the sequential representation. The sequential representation was, however, chosen in order to describe the gradually evolving definitions and the move from requirements to functions to technical solutions to module concepts. It also represents the evolving knowledge of the product.

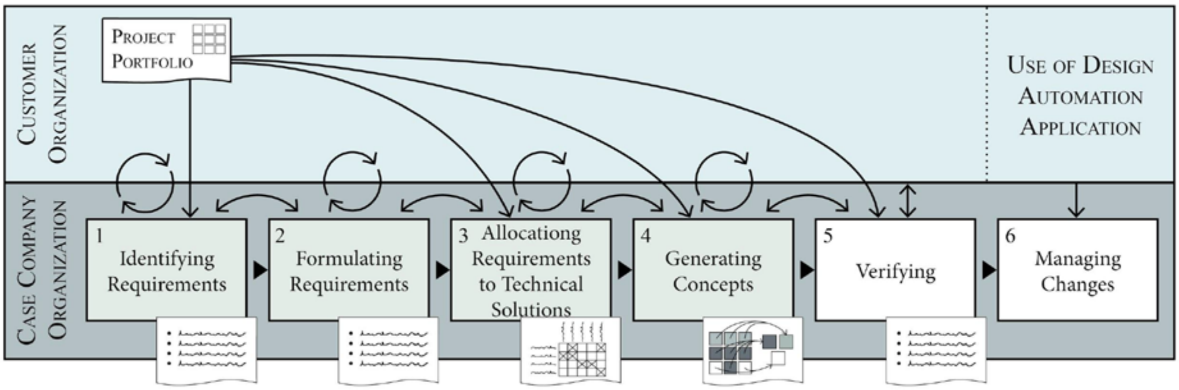

Figure 1 Modularization process for design automation application development of an edge beam, from (Viklund et al 2016).

Identifying requirements
Requirements were identified by customer dialogue regarding the products in combination with an analysis of previous projects in the customer's portfolio. Requirements stemming from the customer were continuously updated as the development process proceeded. This was important since it is difficult for the customer to know how requirements will affect the product, before the results can be visualized. The purpose of the analysis of projects in the customer's portfolio was to find design requirements for the products, to find requirement commonalities on modular or component level. These commonalities were found by studying the variation of technical solutions and to correlate these to their original design requirements. In edge beams, as an example, there are numerous ways of placing and bending reinforcing bars. But if looking from a general perspective, longitudinal bars are meant to deal with transverse loads and brackets are meant to deal with torque forces. So even if all reference projects have different detailed solutions of how to place longitudinal bars and brackets, they fulfill more or less the same requirements. Checklists were used to structure the iterative process.

Formulating requirements
Identified requirements were managed systematically using corporate template files. The interrelations between different requirements were complex. The requirements were managed through document listing the requirements, their interrelations, and their priority. By continuously communicating progressions with the customer, misunderstandings can be kept to a minimum. Identifying and formulating requirements was experienced as a complex undertaking that requires many iterations as knowledge of the product evolves.

Allocating requirements to technical solutions
The process of allocating requirements started at a general level and gradually moved into more detail. There was a constant iteration of analyzing the products function, requirements and the construction solution. For example, the function of the edge beam itself is to distribute loads into the ground. On a more detailed level, insulation solutions can be traced back to the function of thermally insulating the foundation. The variety of technical solutions used in the customer's portfolio to meet these requirements was explored in this step. Lists and matrices were used to find relations between requirements and technical solutions, both for modules and components. Different technical solutions were compared against each other regarding their ability to meets the requirements. Matrices were used to structure the analysis of relations between requirements and technical solutions.

Developing concepts
One of the challenges was to develop parametric technical solutions with a built-in variability. In order to ease the work process Excel spreadsheets were used to structure the design problem and find correlations between modules and components parts, their variability, and defined requirements. AutoCAD was used in parallel to visualize possible solutions and

explore the design field. The visualizations help the engineers to understand the scope of their suggested designs, thus making it possible to combine solutions to decrease the amount of module variants.

Verifying

The platform's variability, both internally within the modules and externally when combining the defined module variants, was evaluated by testing a preliminary version of the design automation application against a variety of previous projects. If all these projects could be solved with the design automation application, the variability of the modules was sufficient. If not, design iteration was made to further develop the module definitions or, if there is limited value from including the outliers, a redefinition was made regarding the scope of the modularized product. The design automation application was also verified through customer tests, based on a well-structured corporate template file. Feedback from this process was used to improve the module definitions and the design automation application itself.

Managing changes

When the design automation application was verified and launched, continuous improvements are made to ensure an up-to-date and well-functioning tool. These improvements are based on feedback from use of the design automation application, including feedback from production. This feedback was used to improve both the design automation application itself and the modular product structure defined within the tool. This last process step focused more on continuous improvements than on final adjustments, thus providing a real-life perspective on the modularized product.

Step 3 and 4 were to a large extent performed in parallel. The outcomes were different though, having a configuration concept as the outcome of Step 4 while Step 3 provided a functional decomposition and a configuration logic. Frequent iterations were also made to Step 1 and 2. The amount of product platform flexibility was defined based on a return of investment strategy of the development process of the design automation application. In order to ensure that requirements regarding e.g. load capacity or deformations are fulfilled, analysis software was used parallel to the development process. Module variance was included in these analyses. These analyses are documented in order to enable knowledge traceability. Constraints and variant opportunities of the configuration concept were continuously communicated with the customer.

3.2 Outer non-bearing wall configurator

Smiding et al. (2016) presented a case when the company developed a product configurator for a configurable platform for outer non-bearing walls. The aim of the broader project was to design and construct a low-energy building suitable for arctic climates. For the wall elements the goal was cost-efficiency, minimizing the transfer of energy through the wall but still being sufficiently flexible to be produced in any production plant or on-site. The development work consisted of two major and relatively separated steps; (A) development of the product structure and (C) programming of the configurator, and a connection step (B) modelling of the modules. Table 1 summaries the development steps.

Table 1 Summary of the design automation development process for outer non-bearing walls from (Smiding et al 2016)

Step	Work activity
A	Business and product evaluation
A	Analyze customers' needs, functional requirements and technical solutions
A	Decomposition of the generic product platform structure
A	Evaluation of the platform's requirements fulfillment
A	Module capability determination and development
B	Modelling of the modules in 3D-CAD

C	Programming the product configuration
C	Testing code and models
C	Creating configurable 2D-models (blueprints)
C	Supporting documents and user interface
C	Overall quality controls

Step A – The development of the generic product structure
(1) Evaluating candidate product types and their possibility for return of investment
(2) Analyzing many different and realized projects blueprints to capture and evaluate market needs and requirements. Analyze and determine the variation in the requirements and technical solutions.
(3) Decompose the generic product architecture into modules and parts, i.e. categorize the suggested technical solutions into modules.
(4) At conceptual level, analyze whether the suggested technical solutions can satisfy the requirements identified in the previous steps.
(5) Develop and determine the flexibility, capabilities and constraints of each module and part.

Step B – Modelling of the modules
(6) Each module is modelled in 3D in CAD software and correlated to their flexibility and capabilities.

Step C – Programming activities
(7) Programming/implementing the modules rules and constraints in the configurator software
(8) Testing and controlling the code
(9) Creation of 2D-models based on the 3D-model and the rules, i.e. mapping and integration of the different software programs
(10) Creation of supporting documents and user interface.
(11) Overall quality control including both product structure, technical solution and code.

Smiding et al. (2016) emphasized that many of the steps interacts and the work is performed throughout iterations between the activities. The study revealed that the work of modularization and programming the configuration was separated but affected each other.

3.3 End frame bridge superstructure configurator

A development process for bridge configurators was formalized through an end frame bridge case where the superstructure was in focus, described in (Sandberg et al., 2016). It has focus on the steps preceding the actual implementation of the configurator and especially the analysis procedures for reinforcement solutions. The process contains 10 steps and is presented in Figure 2.

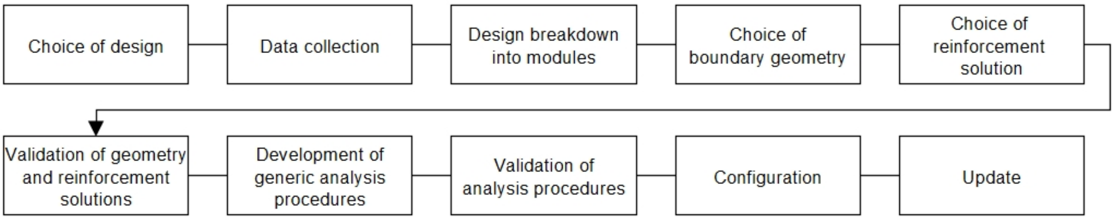

Figure 2 Design automation development process for end frame bridge superstructure adapted from (Sandberg et al 2016).

Choice of design

This step is based on the 1) commonality of the design type, 2) the variation in geometry and 3) the complexity of the dimensioning. By analyzing these three parts the profitability of a possible configurator can be assessed. This is also dependent on the size of the automation project and should be balanced with the return of investment.

Data collection

A data collection is conducted to enable a top-down analysis of the design to enable efficient modularization. From the top-down analysis the design requirements are found by analyzing existing designs combined with the respondent's answers and books regarding design, production and aesthetics. A data collection should produce the following: 1) drawings and documents of existing designs, 2) possible construction methods, 3) aesthetic requirements, 4) experience feedback from production and 5) experience feedback from operation.

Design breakdown into modules

To be able to break down or divide the design into different modules it is important to have knowledge about design, production and aesthetics but also how the drawings are designed and delivered to the customer. This is important because the result should be parameterized geometry that should fulfil all requirements. The design breakdown starts with an analysis of the collected data. The analysis considers: What generates customer value? What requirements are realistic? What functions are associated with the design? What modules can the design be divided into? How do the module interfaces look like? What geometries can fulfil the requirements? How do the geometries affect the building process? How do the geometries affect the operation? What limitations can the reinforcement solutions have on the geometry? By using the collected previous designs, it was possible in a reverse engineering manner to identify starting requirements.

The analysis of how the reinforcement solutions can affect the operation of the bridge needs to be done in four steps: 1) Connect requirements to the design, 2) Divide the design in modules that each fulfil the requirements, 3) Analyze the geometric variation within a module and connect the geometry to the requirements and 4) Analyze that the reinforcement solutions match the geometry.

Choice of boundary geometry

The choice of geometry is based on the analysis of the existing designs and earlier experience of the design. Aspects to consider are: 1) Customer value, 2) aesthetic requirements, 3) other functional requirements, 4) identification of relationships between parameters to decrease the number of parameters, 5) how to facilitate reinforcement solutions, 6) how to facilitate production, and 7) connecting design parts.

Choice of reinforcement solutions

Reinforcement solutions should be chosen so: 1) there is a possibility to fulfil the geometric variation, 2) programming of the dimensioning solutions is simplified, 3) production is possible and 4) material volume is reduced.

Validation of geometry and reinforcement solutions

The geometry choices and reinforcement solutions need to be validated by a group of people with expertise within dimensioning, production and operation of the specific design. This validation can cause iterations where the geometry and the solutions need to be changed and evaluated again. The evaluation has two main purposes: 1) make sure that all functional requirements are fulfilled and 2) make sure that the solutions are done in the best possible way to facilitate design, production and operation.

Development of generic analysis procedures

The analysis procedure should be developed so the input data parameters can be changed to fulfil the functional requirements. It is important to find a balance between the number of input parameters and the variability of the module. By studying existing analysis procedures for different design it is possible to identify common part and reuse them. The different parts can also be identified.

It is important to choose a software that facilitates reuse. The analysis parts should be able to be presented in a way that is easy to evaluate. It should also be easy to extract parts of the analysis to enable reuse in other projects.

Validation of analysis procedures

To make sure that the analysis procedures are correct they need to be tested with different examples and then evaluated. This is done by creating obligatory check boxes that in a clear way show if specific dimensions are approved and letting someone outside the project evaluate the analysis procedure.

Configuration

During this part the configurator is developed. Some important aspects to consider: 1) Clear boundaries for the product parameters should be defined, 2) the GUI for inputting values for the parameters should be user-friendly, 3) the configurator should fit in the current work process and 4) maintenance should be simple. In Figure 3 an example of the 3D geometry and reinforcement solutions is shown.

Updating

To make sure that the configurator will be used in as many projects as possible it is important that the configurator is continuously updated and developed. Updates need to be done simultaneously with every use of the configurator and collection of feedback need to be done after every major stage: design, production and operation.

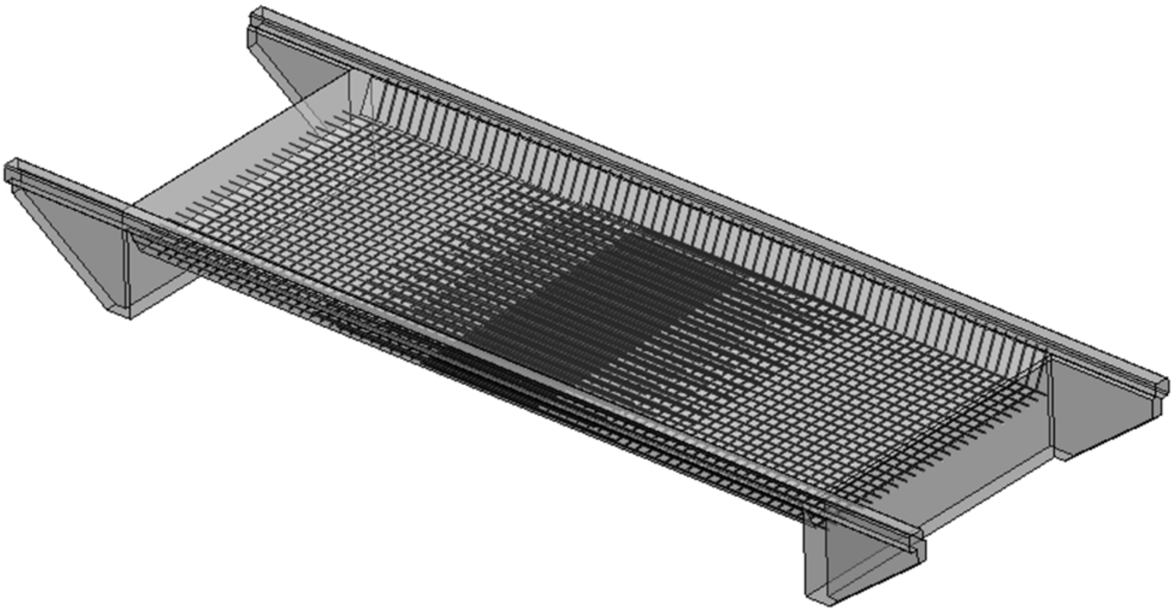

Figure 3 Example of resulting bridge super structure with reinforcement solutions.

4 Proposed general process

In Figure 4 a combination of all three processes is presented. Although the steps are presented in a sequential fashion, iterations between neighboring steps are needed. All three processes have similarities but also specializations. For example, the edge beam process has a requirements focus why all steps of the Requirements lane are from the edge beam process except the first step which is from the bridge process. In the same way the bridge process has focus on the analyses and therefore provides all steps for the Analyses lane. These steps have been slightly renamed to make them more generic, e.g. not all building elements need reinforcement solutions but most need detailed solutions that often need to be guided by analyses. A curtain wall may need energy analyses. The wall process had a focus on the actual programming of the configurator and therefore provide all steps for this lane. The Modules lane has the “Developing product architecture” from the wall process and the “Design breakdown into modules” and “Choice of boundary geometry” from the bridge process. The edge beam’s process step of Generating concepts is integrated and refined in the three steps of the Modules lane.

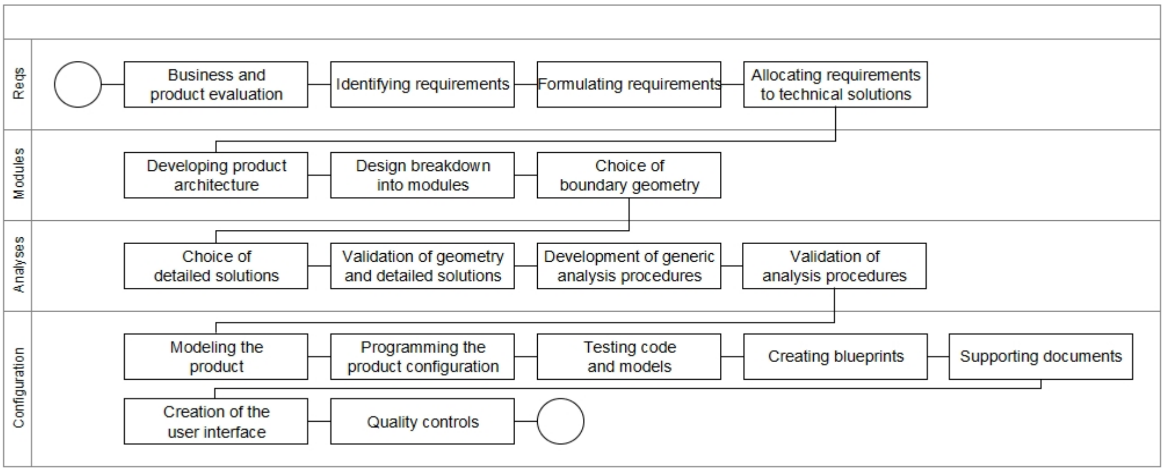

Figure 4 All three processes combined into a general process.

In the Requirement lane, activities related to finding requirements and their relations to each other and to technical solutions are specified. Realized projects in combination with customer dialogue can be used to determine the intended scope of the modularized product. A systematic, yet flexible approach to documentation and analysis is recommended to ensure knowledge capture and traceability. The Module lane consists of the process steps that are related to module decomposition. The logics of how to decompose a product into modules are developed already during the previous process step, “Allocating requirements to technical solutions”. As a result of this activity category, a module-based design concept is generated, ready for design development. This design concept is, at a conceptual level, analyzed against the previously identified requirements. In the Analyses lane, detailed technical solutions are developed and thorough product analyses are performed. These analyses include calculations regarding structural aspects as well as aspects such as energy or moisture transfer, depending on what is relevant to the modularized product. The product is also analyzed to ease programming of the design automation application, and to facilitate design, production and operation. Validation is performed by people within the developing organization in collaboration with people from the customer’s organization.

5 Discussion and conclusion

A general process for design automation application development for building and bridge design has been presented. The most important contribution compared to other processes mainly coming from the manufacturing industry, e.g. (Haug & Hvam 2007), (Erixon 1998), (Stokes 2001) is that this process comes from several processes that in a bottom-up fashion has been developed in construction projects. Also the combination of requirements management, analyses and configuration implementation into one process extend the work of e.g. (Jensen 2014). Bridges and buildings may be different but since there are similarities have on component level this combination can still have value. The presented process could help companies to start develop their own design automation applications in collaboration with design automation consultants and the more experienced they become, the higher is the possibility for companies to create their own applications. This process needs to be tested in cases to identify pros and cons as well as more details for its employment.

References

Blecker, T., & Abdelkafi, N. (2006) Complexity and variety in mass customization systems: analysis and recommendations. *Management Decision*. 44 (7). pp. 908-929.

Blecker, T., Abdelkafi, N., Kreutler, G. & Freidrich, G. (2004) *Product Configuration Systems: State of the Art, Conceptualization and Extension*, Munich Personal RePEc Archive.

Elgh, F. (2008) Supporting management and maintenance of manufacturing knowledge in design automation systems. *Advanced Engineering Informatics*. 22. pp. 445-456.

Erixon, G. (1998) *Modular function deployment: A method for product modularization*. PhD Thesis. Royal Institute of Technology.

Gadde, L. E., & Dubois, A. (2010). Partnering in the construction industry—Problems and opportunities. *Journal of Purchasing and Supply Management*, 16(4), 254-263.

Haug, A., & Hvam, L. (2007) The modelling techniques of a documentation system that supports the development and maintenance of product configuration systems. *International Journal of Mass Customization*. 2 (1/2). pp. 1-18.

Haug, A., Hvam, L. & Mortensen, N. H. (2012) Definition and evaluation of product configurator development strategies. *Computers in Industry*. 63. pp. 471-781.

Helo, P. (2006) Product Configuration Analysis with design structure matrix. *Industrial Management & Data Systems*. 106 (7). pp. 997-1011.

Holmqvist, T. K., & Persson, M. L. (2003) Analysis and improvement of product modularization methods: Their ability to deal with complex products. *Systems Engineering*. 6 (3). pp. 195-209.

Hvam, L., Bonev, M., Haug, A., & Mortensen, N. H. (2014) The use of modelling methods for product configuration in industrial applications. *Proc. of the 7th World Conference on Mass Customization, Personalization, and Co-Creation (MCPC 2014)*, Aalborg, Denmark, February 4th-7th. pp. 529-539.

Hvam, L., Mortensen, N. H. & Riis, J. (2008) *Product customization*. Springer-Verlag Berlin and Heidelberg, Leipzig, Germany.

Jensen, P. (2014). *Configuration of Platform Architectures in Construction*. PhD Thesis. Luleå University of Technology. Sweden.

Jensen, P., Olofsson, T., Smiding, E., & Gerth, R. (2014). Developing Products in Product Platforms in the AEC Industry. *Computing in Civil and Building Engineering*. 2014. pp. 1062-1069.

Jørgensen, K. A. (2001) Product Configuration - Concepts and Methodology. *Manufacturing information Systems Proc. of the 4th SME International Conference*.

La Rocca, G. (2012), Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design, *Advanced Engineering Informatics*, 26, pp. 159-179.

Lovett, P. J., Ingram, A., & Bancroft, C. N. (2000). Knowledge-based engineering for SMEs - a methodology. *Journal of Material Processing Technology*. 107. pp. 384-389.

Sandberg M., Jensen, P., Ramic, Z. and Simonsson. P. (2016) Knowledge-based bridge design, *Proc. of the 19th IABSE Congress - Challenges in Design and Construction of an Innovative and Sustainable Built Environment*, 21-23 September 2016, Stockholm, Sweden

Sandberg, M., Johnsson, H. and Larsson, T., (2008) Knowledge-based engineering in construction: the prefabricated timber housing case, *Journal of Information Technology in Construction*, 13: 408-420.

Smiding, E., Gerth, R., & Jensen, P (2016) Developing Product Configurators in the AEC Industry. *Proc. of the International Conference on Construction and Real Estate Management (ICCREM 2016)*, Alberta, Canada from September. 29th – October. 1st

Stokes, M. (2001). *Managing engineering knowledge - MOKA: Methodology for knowledge based engineering*. ASME Press.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research policy*, 24(3), 419-440.

Viklund, E., Sandberg, M., Lidelöw, H. & Jansson, G. (2016). Modularization Based on Commonalities in House-Building Requirements. *Proc. of 2016 International Conference on Construction and Real Estate Management (ICCREM 2016)*.