

Integrating Knowledge Objects and e-Books to Support Six Roles in the Design Automation Life-Cycle

Joel JOHANSSON¹ and Morteza POORKIANY
School of Engineering, Jönköping University, Sweden

Abstract. Knowledge objects are used to automate engineering processes in a flexible and scalable way. The focus of knowledge objects has been the automation of engineering design. However, in this paper we present a way to integrate knowledge objects into e-books so that formalized engineering knowledge can be read by any stakeholder and still be automatically applied by the system from the same file. The paper shows how different stakeholders throughout the life-cycle of a design automation system benefit from the knowledge object and e-book integration, how the integration is done and an example where the integration was applied. The example is a real example from industry.

Keywords. Knowledge Based Engineering, Design Automation, Design Rationale, Knowledge Management, e-Book

Introduction

Engineering design has been a target for digitalization and automation for decades. During the the 90s it was referred to as expert systems and during the 00s it was called knowledge based engineering. Today we refer to it as digitalization of engineering design. As the computational power increase and software becomes more compatable and platform independent, the more complex tasks can be automated in engineering design, enabling product personalization to unseen degree.

Engineering tasks may be automated within a certain software application by scrip-programming, or by combining software applications by intefrace programming. The automated tasks are often realized by computer code developed by consultants or enthusiastic engineers at the company. In the latter case, the automated tasks are often more sophisticated and company specific than the former, however less documented and professional in implementation. In this paper we suggest using knowledge objects that are integrated to e-books to increase the quality of the automation and its corresponding documentation. This paper focus on how knowledge objects can be integrated with e-books of the EPUB format and what stakeholders benefit from it. In the first section we describe design automation system stakeholders. The second section describes knowledge objects and e-books. The third section explains how knowledge objects and e-books can be integrated. Finally, section four demonstrates the integration by an industrial example.

¹ Corresponding Author, Email: Joel.Johansson@ju.se.

1. Six roles in design automation

We identify six stakeholders of a piece of formalized and automated product knowledge throughout its life-cycle. Figure 1 shows how these stakeholders interact with the design automation system as they share their knowledge during the development process and how they use the system to create new instances of the product. The formalized and automated product knowledge consists of design rationale, design definition and design automation. Design rationale provides explanations of why an artefact is designed the way it is, including statements (textual, numerical or geometrical), argumentations, and decisions [1]. Design definitions define what the product is and how it is manufactured [2]. Design automation consists of computer routines that automate the formalized knowledge stored in the design rationale and design definition. The following subsections describe the six stakeholder roles.

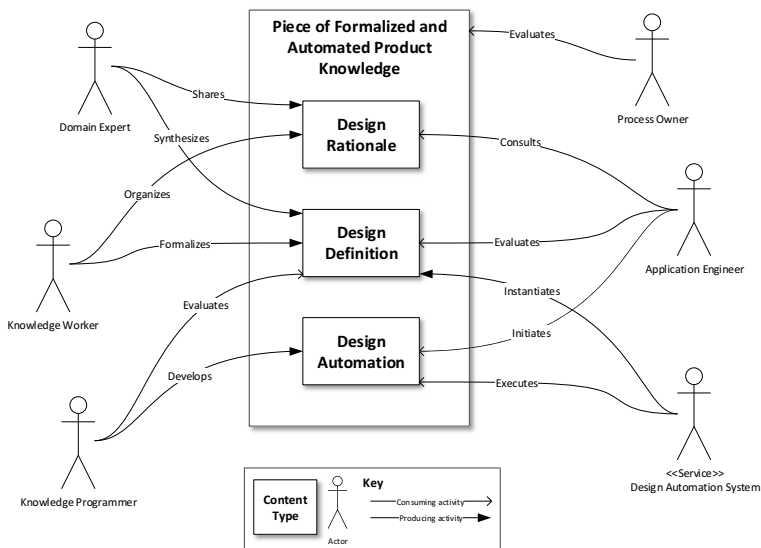


Figure 1. Six roles are identified in the life-cycle of a piece of formalized and automated product knowledge.

1.1. Domain expert

Domain experts are the main source of product knowledge. Their role is to share their knowledge in terms of design definitions and rationale. That means that domain experts provide the organisation (i.e. knowledge workers) with descriptions of the product and underlying technology in any available format including text, sketches, pictures, sound recordings, videos, mathematical formulas, diagrams, three dimensional models, or simulation models. The descriptions should both define the product, i.e. describe how the product is constituted and embodied, and explain why the product is defined that way, i.e. include design rationale. Design rationale consists to a large content of decision history which is described by information about date, decision makers, overall decision objective, decision taken, reason and underlying information for the decision.

1.2. Knowledge worker

Knowledge workers are responsible to harvest knowledge from domain experts through interviews and other knowledge creating processes. They review and organize decisions taken by domain experts which includes categorization and completion of data. The knowledge worker also formalizes design definitions provided by domain experts which includes reviewing documents, mathematical formulas, CAD-models and simulation models. Knowledge workers need to have competence within the domain to some degree but are also skilled in pedagogic and didactics. The knowledge workers should be excellent writers and very communicative. Knowledge workers produce much of the human readable content used to share the knowledge from domain experts and the work is similar to journalists' work.

1.3. Knowledge Programmer

Knowledge programmers automates the formalized and structured product knowledge. They are computer programmers specially trained in knowledge based engineering and artificial intelligence and have good skills in software typically used within the domain, e.g. CAD-systems, FEM-simulations and CAM-systems. Knowledge programmers in the mechanical engineering domain must also be good mathematicians with focus on geometry and topology.

1.4. Application engineer

Application engineers are end consumers of the domain knowledge. They need to have competence enough within the domain to apply the knowledge properly by use of the design automation system. That includes providing the design automation system with enough information so that it can execute automation routines containing formalized and automated knowledge. Application engineer also have the competence to evaluate the design proposal outputted from the design automation system. Application engineers take responsibly for the final design.

1.5. Design automation system

The design automation system schedules the execution of computer routines that have been developed to automate engineering activities in the domain. It needs correct type of information and it needs the information to be accurate and specified in very precise ways. When needed information is provided properly, the design automation can proceed to run the routines and may halt for interaction by application engineers.

1.6. Engineering process owner

The process owner is interested in overviewing the automated process, its effectiveness and efficiency. The engineering process owner may not have competence within the domain, however has a holistic view and perspective on the automated processes and its context.

2. Technological Enablers

In this section we describe two technologies that support the design automation process as outlined in the introduction. The first technology is knowledge objects, which enables flexible and scalable design automation. The other technology is electronic publications.

2.1. Knowledge objects

Knowledge change over time and a knowledge base needs to be flexible so that pieces of knowledge can be easily added, updated, or deleted without disrupting the operation of the system. To make a system flexible in that sense, all parts of the system should be autonomous. Hence, to make the knowledge base flexible, all the chunks of knowledge must be as autonomous as possible. A procedural programming approach is not practical in this case, since changing the knowledge base would mean changing, compiling, and subsequently distributing the programming code. Instead, a declarative way of implementing the knowledge-base has proven to be fruitful.

In a declarative system, the knowledge-base is separated from the functions that make use of the knowledge-base. Consequently, changing the knowledge-base in a declarative system can be performed in a plug-and-play manner. Since the knowledge that engineers use when designing products and developing production tooling is highly connected to various tasks and concepts, the use of object-oriented knowledge bases has proven to be successful. Object-oriented programming offers the possibility to develop highly flexible software. The main idea with objects in object oriented programming is to store related data and functions together. Knowledge objects has been developed to enable flexible and scalable design automation based on object orientation and declarative programming as described in [3–6].

Knowledge object has been defined to be: “bundles of human comprehensible knowledge representations and computer routines for the automated application of the represented knowledge” [2]. To make it possible to develop such bundles, not only has class diagrams been developed but also full functioning software including an inference engine and a knowledge base based on the knowledge object concept. The focus for that development has so far been the automation part but in this paper, we extend it to the description part as well.

Knowledge objects are stored in a knowledge domain which also contains parameters and methods. In [Figure 2](#) an example knowledge domain is shown. In that knowledge domain we have only drawn three connections. Parameter 1 is input parameter to knowledge object 1, parameter 2 is output from knowledge object 2 and method 1 is used by knowledge object 1 transform input to output. As seen in the figure the knowledge domain can have many knowledge objects, parameters and methods and the number of connections can be large.

2.2. EPUB - Electronic publications

EPUB 3.0 is a file format for electronic books that was developed by International Digital Publishing Forum (IDPF) in 2011. There exist free and commercial EPUB readers for all platforms. Several web browsers are capable of reading EPUB as well. EPUB is based on open publication structure, open packaging format and open container format. Roughly speaking, an EPUB can be said to be a web site packed as a single file containing all the web pages as xhtml files, layout information as style sheets, media content as any

standard picture or movie format, table of content as an xml file, and front matter as a picture.

Figure 3 shows the minimum content of an electronic book in EPUB format.

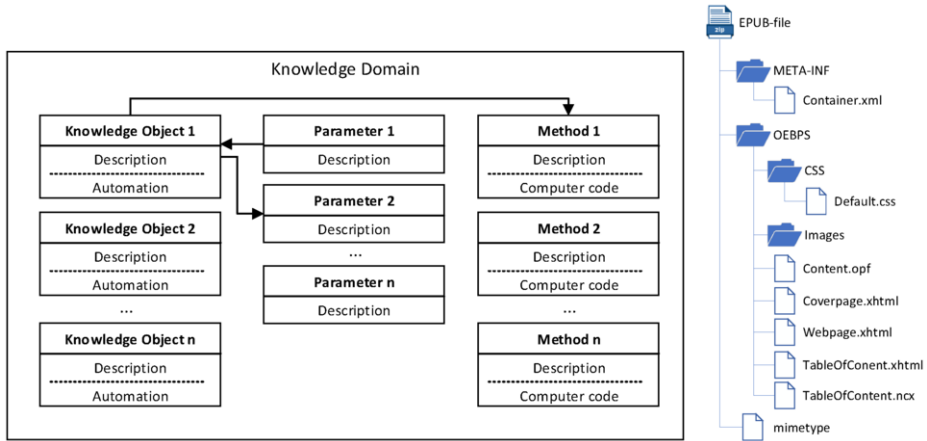


Figure 2. Knowledge domain with knowledge objects, parameters and methods.

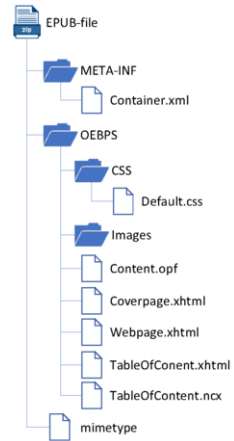


Figure 3: Minimal EPUB file.

3. Supporting the design automation development and maintenance

The use of knowledge objects as a base for design automation has been described in [7,8]. It is here suggested to use the EPUB format to enable flexible and accessible documentation of knowledge objects. In design automation systems based on knowledge objects, it is the knowledge domain that contains all knowledge objects and associated parameters and execution methods, see left side in Figure 4 (which is the same knowledge domain as described in Figure 2). The knowledge domain has a title, a front picture, and a description of the automated engineering process it contains. The description may contain text, pictures and movies. The title and the front picture together form the cover page of the e-book. The description is stored as an xhtml web page with references to pictures and movies in the Images folder.

Each knowledge object in the knowledge domain also contains descriptions regarding the piece of knowledge it automates. The description can be viewed as a formalization and externalization of the knowledge and the attached computer code is the automation of that knowledge. The description consists of text, pictures and movies and is stored as an xhtml webpage. Each knowledge object refers to parameters in the knowledge domain as input parameters and output parameters. The knowledge object also refers to a computer routine. References are made in the webpage for the knowledge object to corresponding webpages parameters and methods.

Each parameter and each method in the knowledge domain contain a description that consists of text, pictures and movies. The description is stored as an xhtml webpage. References to webpages for knowledge objects that use the parameter as input or output are made in the webpage, see bidirectional arrows to the right in Figure 4. These links correspond to the real connections in the knowledge domain making it possible for a reader of the e-book to follow potential execution paths.

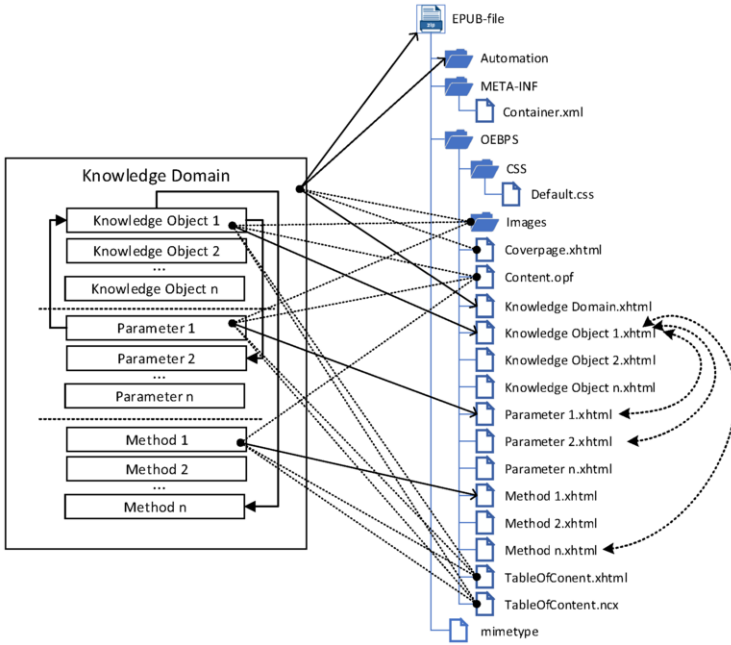


Figure 4. The knowledge domain is integrated with electronic book.

The spine of an e-book in EPUB format is the Content.opf file. That file contains meta-data about the book and a list of its content, what type the content is of, and how the content is referenced in the book. When applied to knowledge objects, the Content.opf file contains the following meta data: title, author name (process owner), unique identifier, language and last modification date. The meta-data is automatically updated when changing the content of the knowledge base which in an efficient way facilitates revision control based on the unique number in the EPUB and its modification date. Each xhtml webpage derived from the knowledge domain, knowledge objects, parameters and methods are listed as chapters. The cover page is of course listed as covers page.

Two files in a EPUB file controls the table of content. TableOfContent.ncx file controls what is listed in the table of content and the TableOfContent.xhtml displays the table of content. It is here suggested to use DSMs (design structure matrix) to order the table of content. That gives a natural flow of the documentation so that knowledge objects early in the process are listed first. The list of parameters and methods are resolved in the same way and will result in a table of content similar to what is shown to the left in in [Figure 4](#).

A final step to entirely integrate the knowledge base with the e-book is to put the knowledge domain with knowledge objects, parameters and methods into an automation folder in the e-book file. When doing so the e-book and the knowledge base becomes one so that the same file is read by humans and interpreted by the machine. However, not all e-book readers accept e-books to contain unidentified content.

4. Example

The case selected to demonstrate the design automation process involves the automated development of toolsets for sheet metal forming. Specifically, a bracket made of high performing sheet metal was selected. The bracket is used in car roof racks. Hundreds of brackets are developed to adapt the roof rack product to almost all car models introduced on the global market. To cut lead time the process of developing the tooling CAD-models was automated. Figure 5 shows the bracket (to the left) which is input to the process, two tooling components (centre) which are the main output from the process, and (to the right) 2D-drawings, PPAP and other documents that are secondary output from the process and not considered in this paper.

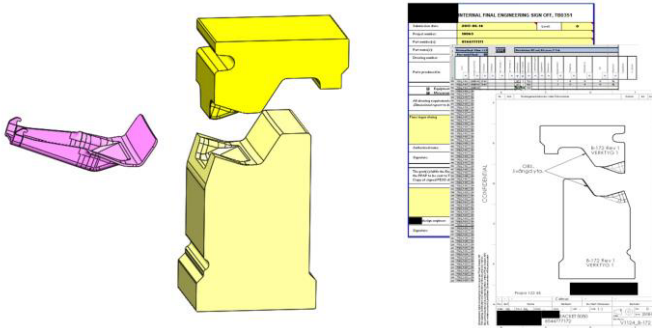


Figure 5. The example system automates the development process of tooling for roof rack brackets [9].

The process of externalizing, formalizing and developing a prototype system for the design automation has been described in [9]. After that work, the prototype system was reshaped into knowledge objects. The final automation that was based on knowledge objects consists of 22 knowledge objects and 51 parameters. The process is executed interactively by the application engineer from the CAD-system and assumes a CAD-model of the roof rack bracket to be active. The process is: 1) Retrieve parameters from the CAD-model. 2) Construct surfaces in the bracket model to be used as tooling surfaces. 3) Create an instance of correct tooling CAD-template and put in folder as a new tooling model. 4) Update parameters in the new tooling model. 5) Add geometry in the new tooling model and remove tooling surfaces that were constructed in step 2. In the following subsections, we describe how each stakeholder produce and consume knowledge throughout the life cycle of the example design automation system.

4.1. Domain expert

Two domain experts have shared their knowledge in the example case. The first one is a senior tooling engineer with 35 years of experience. The knowledge he provided is general for how the tooling should be constituted and defines the design intent of each component in the tool. The other domain expert is an engineer with some 8 years of experience of tooling development. He has been working with the first expert and has learned from him but also got his own experience through projects during the years. This domain expert has detailed knowledge regarding all variants of components and corresponding CAD-models and other documents.

System view: The domain experts view the design automation from the e-book. They are interested in the detailed descriptions of knowledge objects and parameters.

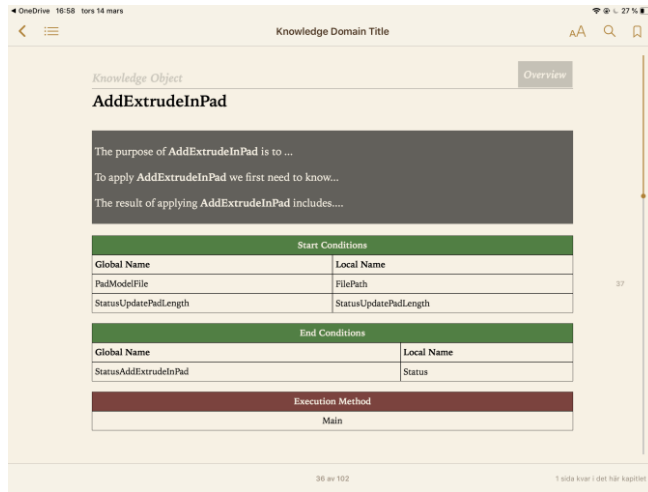


Figure 6. Screen shot of the e-book opened in IBooks on an Ipad.

4.2. Knowledge worker

Two knowledge workers are identified in the example case. The first one is the researcher that interviewed the domain experts to externalize and formalize their knowledge. The second one is the researcher that restructured the formalized knowledge into knowledge objects and did detailed descriptions of it.

System view: The knowledge workers view the design automation from the Howtotation® user interface, see Figure 7, that consists of an interactive process overview, a description pane to review descriptions (currently it cannot be edited from the user interface, but it is planned to be added). The user interface allows the knowledge workers to manipulate the knowledge base by adding/editing/removing knowledge objects and parameters. The focus for the knowledge workers is the descriptions of knowledge objects and parameters.

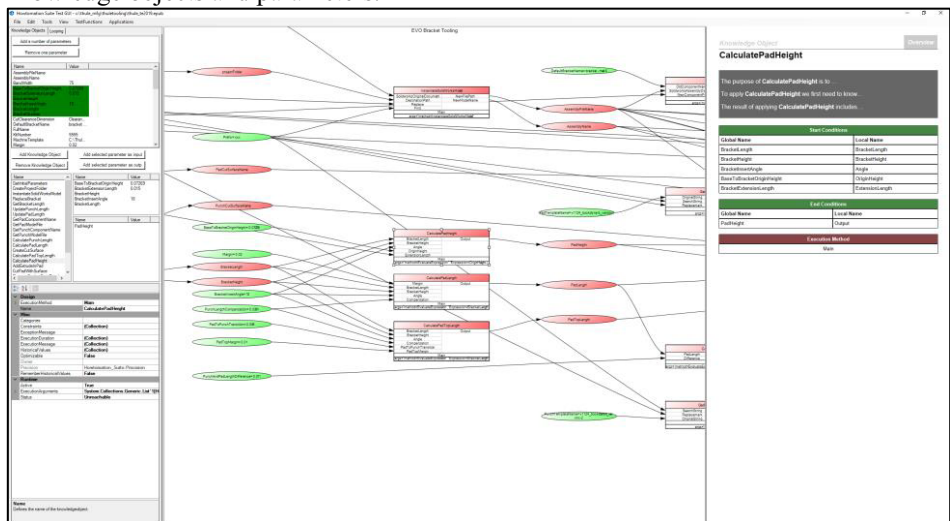


Figure 7. Screen shot of the Howtotation® user interface.

4.3. Knowledge Programmer

Two knowledge programmers are identified in the example case. The first one is the researcher that developed the prototype system for automating the tool development process. The second one is the researcher that reshaped the prototype system into a knowledge object based design automation system.

System view: The knowledge programmers also view the design automation from the Howtotation® user interface. The focus for the programmers is to develop and add methods to the system. To support the programmers, it is possible from the Howtotation® user interface to run the system to follow its procedures for debugging purposes. Knowledge objects and parameters highlight in green when successfully executed.

4.4. Application engineer

There is one application engineer identified, which is the same person as identified as the second domain expert. He develops tools for each new bracket variant. It has been a painstaking work that now has been automated. The automation enables a new less experienced tooling engineer to perform that task.

System view: The application engineer sees the design automation system as a button or menu option in the CAD-system. He or she may not be aware of how the automation is performed or any of the underlying knowledge.

4.5. Design automation system

The automation was done in the Howtotation® software, developed by one of the authors of this paper. The software has been presented before in this conference [8,10] and has now been extended to support the documentation through e-books.

System view: the heart of the design automation system is the inference engine. In the example system the inference engine is tailored to handle knowledge objects.

4.6. Engineering process owner

The owner of the design automation process is the owner of the tooling development process. In the example case that is a chief engineer that oversees all tooling development at the company.

System view: The process owner views the design automation from the e-book and is especially interested in the process overview and description of the knowledge domain.

5. Conclusion

This paper describes how six roles in the life-cycle of a design automation are supported by knowledge objects and electronic books. Knowledge objects carry formalized knowledge and corresponding computer routines to automatically apply the knowledge. The formalized knowledge is distributed through autogenerated e-books that can be

accessed and read regardless of platform. This way of automating engineering processes makes the resulting system modularized and flexible to changes but also well documented even after change which in turn makes them live longer. It is possible to fully integrate e-books and knowledge based engineering based on knowledge objects so that it is the same file that is read by humans and interpreted by the machine. The benefits are that the automated engineering processes are transparent and can be maintained over time, better revision control due to unique identifiers in the EPUB format, knowledge management and design automation is done in one system so that documentation is updated when the knowledge base is changed.

Focus of the work presented in this paper has been to develop the infrastructure of sharing formalized and automated product knowledge as knowledge objects integrated in with e-book. Future work includes extended verification and validation of the proposed method.

Acknowledgement

The research was performed within the research ProAct funded by the Jönköping County, Sweden. The authors also want to express their gratitude to the engineers and other personnel at the company where the case study was performed.

References

- [1] M. Poorkiany, J. Johansson and F. Elgh, Capturing, structuring and accessing design rationale in integrated product design and manufacturing processes, *Adv. Eng. Informatics*. **30** (2016) 522–536. doi:10.1016/j.aei.2016.06.004.
- [2] J. Johansson and F. Elgh, Knowledge Objects Enable Mass-Individualization, in E. Andrés-Pérez, L.M. González, J. Periaux, N.R. Gauger, D. Quagliarella and K.C. Giannakoglou (eds.) *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, 2019, Springer International Publishing, pp. 371–386.
- [3] J. Johansson, A flexible design automation system for toolsets for the rotary draw bending of aluminium tubes, *2007 Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf. DETC2007*. **4** (2008) 861–870.
- [4] J. Johansson, How to Build Flexible Design Automation Systems for Manufacturability Analysis of the Draw Bending of Aluminum Profiles, *J. Manuf. Sci. Eng.* **133** (2011) 061027. doi:10.1115/1.4005355.
- [5] F. Elgh, and J. Johansson, Knowledge object - A concept for task modelling supporting design automation, *Advances in Transdisciplinary Engineering*, Vol. 1, IOS Press, Amsterdam, 2014, pp. 192–203. doi:10.3233/978-1-61499-440-4-192.
- [6] J. Johansson and F. Elgh, How to successfully implement automated engineering design systems: Reviewing four case studies, in: *Proceedings of the 20th ISPE Int. Conf. Concurr. Eng. CE 2013, 2013*, pp. 173–182. doi:10.3233/978-1-61499-302-5-173.
- [7] V.K. Rao Pabolu, R. Stolt and J. Johansson, Flexible manufacturability analysis applied to the welding process to increase sustainability in engineer to order businesses, *Int. J. Agile Systems and Management*, **10** (2017), Issue 3–4, <https://doi.org/10.1504/IJASM.2017.088536>.
- [8] J. Johansson, Howtomatic© Suite: A Novel Tool for Flexible Design Automation. *Advances in Transdisciplinary Engineering*, Vol. 2, IOS Press, Amsterdam, 2015, pp. 327–336.
- [9] T. Heikkinen, J. Johansson and F. Elgh, Extended design assets enabling automated tool development as a part of a product platform approach, *15th Int. Des. Conf. - Des. 2018, May 21st-24th, 2018, Dubrovnik, Croat.* (2018) 757–768. <http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-39681>.
- [10] V.K.R. Pabolu, R. Stolt, and J. Johansson, Manufacturability analysis for welding - A case study using howtomatic© suite, *Advances in Transdisciplinary Engineering*, Vol. 4, 2016. doi:10.3233/978-1-61499-703-0-695.