20th ISPE International Conference on Concurrent Engineering
C. Bil et al. (Eds.)

doi:10.3233/978-1-61499-302-5-173

How to Successfully Implement Automated Engineering Design Systems: Reviewing Four Case Studies

JOEL JOHANSSON, FREDRIK ELGH
Mechanical Engineering, School of Engineering, Jönköping University, Sweden

Abstract. Introducing computerized systems to automate engineering design activities (design automation) on manufacturing companies promises improved product quality, shortened time to production, increased control on cost, less effort to adapt products to new customer requirements. Due to these motives, big effort has been put on developing computer systems automating a variety of engineering design activities throughout the product and production development process. The question is now: Is design automation ready to launch yet? In this paper, we review four cases of design automation of engineer-to-order to give guidelines for developing engineering design automation systems.

Keywords. Automated Engineering Design, Knowledge Object, Inference engine

1. Introduction

Introducing computerized systems to automate engineering design activities (design automation) on manufacturing companies promises improved product quality, shortened time to production, increased costing control, and less effort to adapt products to new customer requirements. Due to these motives, big efforts has been put on developing computer systems automating a variety of engineering design activities throughout the product and production development process. Here, we only consider the design automation of products that are engineered-to-order, i.e. products for which it is not possible to develop pre-defined sets of configurations to select from.

To develop guidelines for selection of technical solutions four design automation projects (driven by the authors during the 2002-2012) are reviewed in this paper. The systems were rated according to defined criteria for good design automation systems. The scores were then compared to the content of the system (the automated engineering knowledge) in order to see how it affects the resulting systems. The systems were evaluated based on the following criteria [1]:

Transparency: The level of clearness and accessibility of the documentation and visualization of, the product and its product structure, design process, design tasks, and design knowledge. A highly transparent process is to be seen as an antonym to a black-box process.

User readable and understandable knowledge: The level of which the knowledge (e.g. design rules) is expressed in a user readable and understandable format. Rules expressed in formal language are for example easier to read and understand, but less efficient in execution, than rules expressed in some computer programming language.

Scalability: The possibility of expanding the system towards higher system complexity through system realization architecture that allows the application to grow and expand with emerging details, additional or refined tasks to be performed, additional knowledge to be added, and additional application modules to be implemented.

Flexibility: The possibility of expanding the system within the same level of system complexity through realization architecture that allows the application to grow and expand with additional variants, products, and/or sub-systems.

Longevity: Factors that can affect system longevity are for example: dependence on single specialized vendors, level of transparency, level of user readable and understandable knowledge, and ease of application overview and maintenance.

Investment: Level of initial cost for implementation and use of capital recourses (in relation to a predicted total cost of system development and operation).

Effort of development: Level of development (and expansion) effort in terms of the use of human resources.

The following systems are detailed and evaluated in the paper:

The CoRPP system (2003): This system was developed as a research case study and targets the preliminary design of a bulkhead part of a submarine escape section for subsequent cost calculation [2].

The Kongsberg-Automotive system (2007): This system automated layout of heating elements for car seats in order to make cost calculation and production planning [3].

The BendIT system (2008): This system targets the development of toolsets for the rotary draw bending of aluminum profiles. That system combined KBE, CAD, and FEM to make design proposals of the tools to subsequently simulate and analyze production outcome .

The TRackS (2010): This system targets the development of ski-rack for cars with no rails. Specifically the system is used to retrieve existing components to make new combinations for new car models. When the combinations of components are established the behavior of the complete ski-racks during car collision are automatically simulated using FEM-simulations [4].

2. System descriptions

This study is based on four cases, which are briefed in this section. All the systems were developed as parts of the research projects, as is common for many automated design systems targeting products that are engineered to order.

2.1. System 1: CoRPP – Knowledge processing static flow

The primary purpose of the CoRPP (Coordinated Realisation of Products and Processes) system was to support the company in its effort to gain design solutions with enhanced producibility through studies of variations in cost, weight and operation time.

The main element of the bulkhead is a circular plate with vertical structural members, which consist of cut, rolled and welded steel plating, as shown in Figure 1.

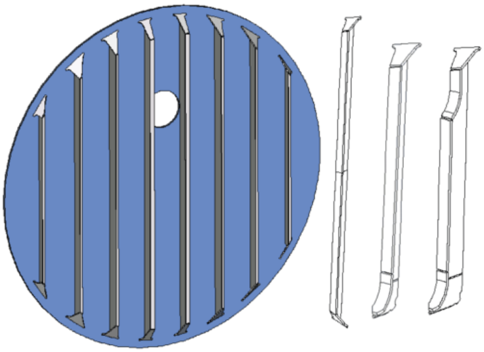

Figure 1. A bulkhead and examples of stiffener variants.

The system architecture was modular, where the knowledge was captured in knowledge objects grouped in separated modules.

The knowledge base (knowledge objects in the different modules) is executed on the basis of different customer specifications. The product design module generates parameters that serve as input to product geometry, process planning and cost estimation. Product geometry, process planning and cost estimation consist of a number of interrelated knowledge objects (generic templates) that are instantiated and then executed and configured in accordance to the input parameters.

The system was developed together with an industrial partner and a research institute using a commercial off-the-shelf (COTS) approach (comprised of MS Access, MS Excel, MS Visual Basic, Mathsoft Mathcad, and PTC Pro/Engineer). The modules for process planning and cost estimation were developed by one of the authors. The system was considered to have many areas of use at the company: design calculations, design optimization, geometry modeling, automated CAD generation, knowledge repository, design manual, process planning, cost estimation, operation time estimation, and weight calculations.

The system consists of a geometry modeler separated from commercial software for solid modeling. An extended product model was implemented in the geometry modeler supporting the process planning and cost estimation of the product.

The bulkhead was modeled in a software application as parametric solid models, using methods that permit dimensional and topological changes [5]. The geometry modeler drives the parametric solid models. A nomenclature was defined and implemented. This enabled the mapping between the geometry modeler and in the standard process plans. Standard process plans, with the integration of a system for cost estimation, were created in a common spreadsheet software application. The operations in the process plans were activated in either of two ways: if there was a corresponding feature in the geometry model, or in accordance with rules where operations are interrelated. Geometrical and topological cost drivers were identified and corresponding parameters stated in the standard process plan. Production data and costs for production resources were gathered in tables.

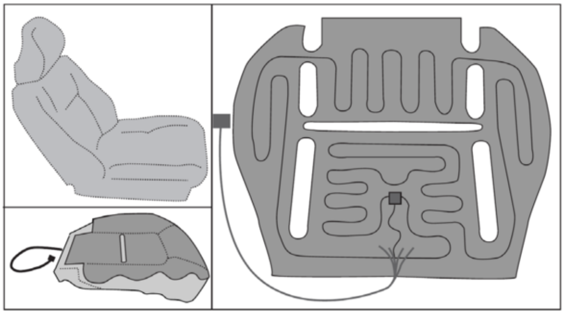

Figure 2: Heat elements

2.2. System 2: KA – knowledge processing static flow and information handling

The scope of the KA-system was to generate variant designs of heating elements based on different customer specifications and seat geometries. The heating elements are part of a car seat heater. The heating element consists of a carrier material, a wire and a connecting cable. The wire is laid out and glued in a pattern of sinusoidal loops between the two layers of carrier (Figure 2).

The pattern is calculated on the basis of company-aggregated knowledge. The purpose was to combine some of the functions and properties relating to information handling and knowledge processing into one system. The objectives with the system were: cut quotation lead-time, allow for evaluation of different design alternatives, quality-ensure the design process, capture design knowledge, ensure producibility, and provide design documentation.

The system was developed by one of the authors in cooperation with programming consultant. The knowledge base comprises rules in Catia Knowledge Ware Advisor (KWA). The rules are linked (through an Access database) to different Knowledge Objects. The Knowledge Objects can be of different types (e.g. Catia KWA rules, Mathcad worksheets) in which the methods of the different Knowledge Object are implemented. The rule firing, invoking the Knowledge Objects, is controlled by an inference engine (CATIA KWA in early versions, and in-house developed in later versions of the system). The company resources with associated manufacturing requirements are stored in an Access database together with the Knowledge Objects. The graphical user interface and the interfaces to different software applications and databases are programmed with Visual Basic. The system is fed with customer-specific input (parameter with associated values together with a 2D outline of the heated seat areas). The main output is the pattern for the heating wire's centerline, an amplitude factor for the sinusoidal loops and the wire specification.

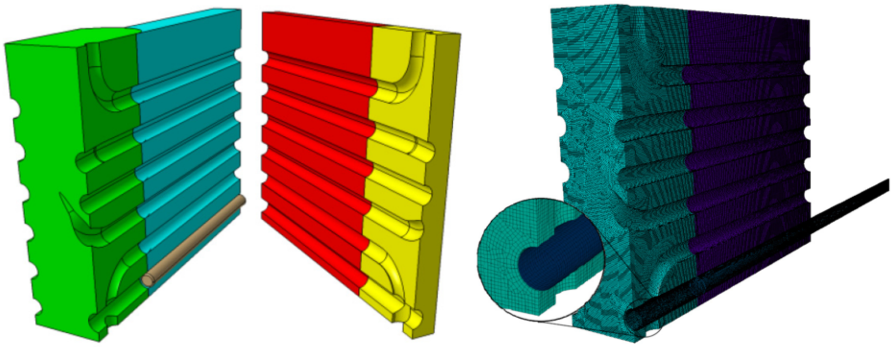

Figure 3: Suggested design as CAD model (left), and analysis model (right) of a rotary draw bending toolset.

2.3. System 3: BendIT, Knowledge processing, dynamic flow

The target for the BendIT system was to design tool-sets for the rotary draw bending of aluminum profiles with general sections. The complete process was fully automated including initial estimations of spring back, required bending moment, need for section support and other phenomena based on handbook formulas and formulas derived from fundamental physical laws to generate a design proposal represented in CAD software (left in Figure 3). To render the CAD-model first the volume allocated by the profile during all the manufacturing steps had to be generated (this was done using automated CAD-functionalities), subsequently template CAD-models of tool-sets were retrieved and the previously generated geometry was removed using boolean operations to have the tool cavities. The design proposal was then used to generate simulation models for each manufacturing step in the tool-set (right in Figure 3). The results from the simulations, the simulated production outcome, were automatically analyzed for wrinkling of the profiles.

The structure of the system was completely modular based on knowledge objects. The solution path of the knowledge base was dynamical so that knowledge objects were executed on demand, controlled by an inference engine developed by one of the authors. The knowledge objects were used to connect to MS Excel, CATIA, MS Access, PTC MathCAD, and LS-Dyna. Additionally, routines were developed and automated through knowledge objects to convert CATIA mesh models to LS-Dyna, to make suggestions on where to support the profiles, and to detect wrinkles.

In the system, it was possible to add redundant knowledge. In other words knowledge based on rules of thumb, knowledge based on formulas analytically derived from fundamental laws of physics, knowledge based on experiments, and knowledge based on simulations, could all exist for same phenomena at the same time. For an example there were three knowledge objects calculating the developed length of a circular aluminum tube. Meta-knowledge was added so that the special inference engine could execute appropriate knowledge objects in different context of running the system.

The system was finally used to investigate the design space of general aluminum profiles.

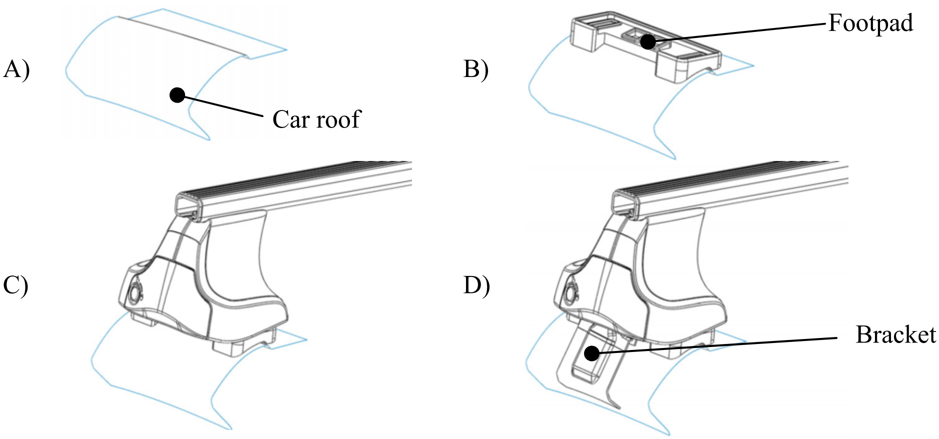

Figure 4: Thule clamp development process.

2.4. System 4: TRackS, information handling and visualizing

The Thule Rack System (TRackS) was developed targeting the automation of adapting a special product to new specifications. The system utilizes the Case Based Reasoning method to retrieve exiting components to assemble into new product variants. The method was applied on two targeted components, one where the search could be performed directly on component geometry and one where the search was based on clearance analyzes.

The system was developed using procedural programming and was embedded as an add-in to the SolidWorks CAD-software. The user creates a new project in the add-in, selects the roof model on which the system makes a new assembly inserting the roof in correct position. The user then creates two datum-planes to indicate where to place the racks on which the system automatically searches for existing footpads that would fit on the roof at the given positions. When finished the search, the user can evaluate footpads based on fast in-context previews. When found, the good fitting footpads are automatically retrieved and placed on the roof in the assembly. Subsequently the racks are mounted on the footpads and a new search procedure for good brackets starts, also including fast in-context previews.

When the rack model is complete, including footpads and brackets, a simulation model is automatically developed in order to do crash simulations. The simulation models are generated using a name convention used in the CAD-files to define how the part should be idealized together with macro programming for the FEM pre-processor (ANSA).

3. System contents

If the design automation system somehow deals with the embodiment of the product, it needs to be capable of geometrical modeling. It is experienced by the authors that introducing geometrical modeling into a design automation system affects criteria 1, 2, 6 and 7 negatively. All the reviewed systems included geometrical modeling (as that is fundamental for design this is what makes the big difference to other computer systems, see Table 1). The geometrical modeling capability can be implemented using either commercially available CAD-systems or in-house developed routines. When using commercial CAD-systems the resulting product geometries can be rendered either by adapting pre-defined template-models that might vary parametrically and topologically, or by generating it by macro programming.

The first system used parameterized CAD-templates adapted to new product specifications while the second system instead generated the geometry. The third system was hybrid using templates adapted through Boolean operations of generated geometry. The last system used the CAD-system for previewing geometry only since the geometrical functions were instead implemented into the systems core code. The main reasons were the lack of the necessary functionality (clearance analyses of footpads and brackets) and that communication through the API made the performance poor.

The engineer-to-order-process often includes the simulation and analysis of the suggested geometry through FEM-calculations. The system would then include automatic idealization and meshing of the product. In addition, boundary conditions, constraints and other definitions have to be generated. The two last systems included the automation of FEM-analyzes based on naming CAD-features and macro programming.

Table 1. Categorizing the knowledge content of the systems.

	Geometry	CAD	In-house	Templates	Generative	FEM
1. CoRPP	X	X		X		
2. KA	X	X			X	
3. SAPA	X	X		X	X	X
4. TRackS	X	X	X	X	X	X

Suggested increased difficulty

4. System structure

The system structure can be either modular or not. If the system is modular, its execution flow can be fixed predefined, runtime static or runtime dynamic. A fixed predefined flow means that the execution sequence is determined during the design of the system and hard coded into the machine code. A runtime static execution flow means that the modules are executed in a predefined order that is editable in the system without rebuilding it. A runtime dynamic flow means that the modules are executed based on current status of the system, either whenever there is enough information (as soon as possible) or on demand (as late as possible).

The first three systems were based on a modular structure called knowledge objects, of which the first used a fixed pre-defined execution flow, the second used an inference mechanism that resolved the execution order when the system is invoked based on the knowledge objects dependencies, and the third system executed the knowledge objects dynamically whenever enough information was present at run-time, see Table 2.

Table 2. Categorizing the structures of the systems.

	Modular	Fixed pre-defined flow	Run-time static flow	Run-time dynamic flow	As soon as possible	On demand
1. CoRPP	X	X				
2. KA	X		X			
3. SAPA	X			X	X	
4. TRackS		X				

5. Evaluation of the systems

The four systems were compared based on the criteria mentioned in the introduction assigned values poor, moderate, and good, see Table 3.

5.1. Transparency

The first three of the systems were based on a modular structure, i.e. knowledge objects. The two first systems were built on knowledge objects automating widespread commercial software making the human readability of the knowledge high whereas the knowledge automated in last system was built into machine code. Even though the third system was modular and built on knowledge objects some of the knowledge chunks was hard coded into machine code, making the readability of the system moderate.

Common for the first three systems also was the usage of the DSM. In the first two systems the design process was visualized using a DSM that provided access to underlying executable rules and documentation (both general and variant specific). In the third system, the dependencies of knowledge objects and/or design parameters were visualized through DSMs.

The product structure in CoRPP was not explicit as in the KA system, which proved to enhance the transparency of the system.

5.2. User readable knowledge:

The knowledge for design calculations was explicit in the first three cases. It was made explicit by wide-spread commercial software for spreadsheets and mathematics. The CAD modeling rationale was treated differently though. In system 1, the geometry calculations were separated from the CAD software which enabled a possibility to add notes on the rationale. However, this was very time consuming and required quite a lot of effort in the development phase. In the second system all geometry was treated in a CAD software. An extensive macro was developed that required a separate documentation for future maintenance. Even though, geometrical calculations were made explicit some of the routines were developed in-house and complied to machine code. In system 4, the underlying knowledge was completely hard coded making it hard to read.

5.3. Scalability:

System 1 and 4 had a hard coded execution sequences and product structures. System 2 had an executions sequence resolved in run-time by an inference mechanism and a flexible product structure. In system 3 the execution of knowledge object was based on current system status. It was possible to add redundant knowledge objects making for a highly scalable and flexible system.

5.4. Flexibility:

System 1 and 4 were special purpose systems while systems 2 and 3 were built using in-house developed platforms for DA-applications that are general and could be applied on a variety of design activities.

5.5. Longevity:

As a result from the above.

5.6. Investment:

The development of systems 1 to 3 included developing platforms for automating engineering design which required quite a lot of man-hours. However, the investments in software were at a minimum. System 4 was targeting the automation task directly, without the development of a platform, and took about one man-year to develop. That system was integrated as an add-in to the already existing CAD-system at the company.

5.7. Effort of development:

The design calculations and the separation of geometry algorithms required quite a lot of effort in the system 1 and 4 system whilst the programming of the CAD-macro and in-house routines for geometry handling was the most time consuming tasks in the development of system 2 and 3.

Table 3. The different systems were scored based on the criteria mentioned in the introduction.

Critera	CoRPP	KA	BendIT	TRackS
1. Transparency	Good	Good	Moderate	Poor
2. User readable knowledge	Moderate	Moderate	Moderate	Poor
3. Scalability	Moderate	Good	Good	Poor
4. Flexibility	Poor	Good	Good	Poor
5. Longevity	Moderate	Good	Good	Poor
6. Investment	Moderate	Moderate	Moderate	Good
7. Effort of development	Moderate	Moderate	Moderate	Moderate

6. Conclusions

When comparing the contents of the systems to the evaluations it seems that, when possible, the use of template-models in commercial CAD-systems to automate embodiment of engineer-to-order products increases the possibility of a successful design automation project. It is also seen that the introduction of automated analysis through FEM-simulations is difficult and affects criteria 1 to 5 negatively.

We can conclude that when planning the automation of engineering design processes it is important to consider what types of content the final system will have. The answers to the following questionings can serve as an indication of how well the result will meet the criteria stated in the introduction:

- Does the system need to interact with geometry?

- Is it possible to utilize commercially available CAD-systems or is there a need for specialized geometrical functions?

- If CAD-software may be used, can the geometries be represented using adaptable templates or should they be generated?

- Is there a need for frequent geometrical evaluation? If so, will the performance of the system when using a CAD-software compared to developing own functions affect the usability of the system?

- Will the system automate FEM-simulations?

The more affirmative answers, the harder it will be to implement a system meeting all the criteria.

Comparing the system structures to the evaluations shows that the system structure should as far as possible be modular and the execution flow should not be fixed but runtime static or runtime dynamic.

Of course, many factors affect the result of a design automation project of which this paper just touches some few. However, since the computerization of engineering design is rapidly increasing further investigation is necessary. It seems for instance hard to decide the size of knowledge objects, how to document them, and how to manage the organization around the design automation system.

Further, the development of and commercialization of software facilitating all the necessary functionality to support the automation of engineering design is demanded.

7. Acknowledgements

The authors express their gratitude to all sponsors funding and participants for technical support in the projects mentioned in this paper.

References

1. Cederfeldt, M., *Planning Design Automation : A Structured Method and Supporting Tools*, in *Institutionen för produkt- och produktionsutveckling, produktutveckling*2007, Chalmers university of technology: Gothenburg.
2. Elgh, F. and M. Cederfeldt, *Concurrent cost estimation as a tool for enhanced producibility-System development and applicability for producibility studies.* International Journal of Production Economics, 2007. **109**(1-2): p. 12-26.
3. Elgh, F., *Decision support in the quotation process of engineered-to-order products.* Advanced Engineering Informatics Advanced Engineering Informatics, 2012. **26**(1): p. 66-79.
4. Johansson, J., *Combining Case Based Reasoning and Shape Matching Based on Clearance Analyzes to Support the Reuse of Components*, 2012.
5. Cederfeldt, M. and S. Sunnersjö, *Solid Modelling with Dimensional and Topological Variability*, in *Research for Practice - Innovations in Products, Processes and Organisations : Proceedings of the 14th International Conference on Engineering Design, August 19 - 21, 2003, Stockholm, Sweden.*2003, The Design Society: Glasgow.