Design Automation Systems for Production Preparation

Applied on the Rotary Draw Bending Process

JOEL JOHANSSON

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2008

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Design Automation Systems for Production Preparation

Applied on the Rotary Draw Bending Process

JOEL JOHANSSON

TEKNISKA HÖGSKOLAN
HÖGSKOLAN I JÖNKÖPING

Department of Mechanical Engineering
SCHOOL OF ENGINEERING
JÖNKÖPING UNIVERSITY
Jönköping, Sweden

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2008

Design Automation Systems for Production Preparation
Applied on the Rotary Draw Bending Process

Joel Johansson

Department of Mechanical Engineering
SCHOOL OF ENGINEERING
JÖNKÖPING UNIVERSITY
SE- 551 11 Jönköping, Sweden
Joel.Johansson@jth.hj.se

Research Series from Chalmers University of Technology
Department of Product and Production Development
ISSN 1652-9243
Report No. 33

Published and Distributed by
Chalmers University of Technology
Department of Product and Production Development
SE – 412 96 Göteborg, Sweden

Printed in Sweden by
Chalmers Reproservice
Göteborg, 2008

ABSTRACT

Intensive competition on the global market puts great pressure on manufacturing companies to develop and produce products that meet requirements from customers and investors. One key factor in meeting these requirements is the efficiency of the product development and the production preparation process. Design automation is a powerful tool to increase efficiency in these two processes.

The benefits of automating the production preparation process are shortened led-time, improved product performance, and ultimately decreased cost. Further, automation is beneficial as it increases the ability to adapt products to new product specifications with production preparations done in few or in a single step. During the automation process, knowledge about the production preparation process is collected and stored in central systems, thus allowing full control over the design of production equipments.

Three main topics are addressed in this thesis: the flexibility of design automation systems, knowledge bases containing conflicting rules, and the automation of the finite element analysis process. These three topics are discussed in connection with the production preparation process of rotary draw bending.

One conclusion drawn from the research is that it is possible to apply the concept of design automation to the production preparation process at different levels of automation depending on characteristics of the implemented knowledge. In order to make design automation systems as flexible as possible, the concept of object orientation should be adapted when building the knowledge base and when building the products geometrical representations. It is possible to automate the process of setting up, running, and interpreting finite element analyses to a great extent and making the automated finite element analysis process a part of the global design automation system.

Keywords: Design automation, Rotary draw bending, Knowledge Based Engineering (KBE), and Finite Element Analysis (FEA).

ACKNOWLEDGEMENTS

The research carried out through this thesis has been conducted in corporation between Jönköping University, Sapa Group, Faurecia, and Ekenäs Mekaniska Verkstad. I am very grateful for the funding by Sapa Group and the Knowledge Foundation (in Swedish KK-stiftelsen). It has allowed me to concentrate on this specific research area that is close to my passion for mechanical engineering and computer programming. I also express my gratitude to Chalmers University of Technology, Department of Product and Production Development and Professor Rikard Söderberg, for kindly accepting me as a PhD student, making this work possible.

I also express my gratitude to my supervisor, Staffan Sunnersjö, for his enthusiasm and belief in my ideas and guidance in my research efforts. I also thank Mikael Cederfeldt, Fredrik Elgh, and Roland Stolt for support and interesting discussions on the subject of design automation and research in general. Niclas Strömberg is also gratefully acknowledged: for doing FEM-calculations and also for guiding me when I performed FEM-calculations.

Finally, I thank my family, especially my beloved wife, Synnöve Johansson, and my friends who have always stood by me, giving me their support and comfort.

Jönköping, 2008

Joel Johansson

PUBLISHED PAPERS

This licentiate thesis is based on the following published and appended papers:

Paper A – J. Johansson and S. Sunnersjö (2006). *Automated Design of Rotary Draw Bending Tools – An Approach Based on Generic CAD-Models Driven by Heuristic and Algorithmic Knowledge.* EDMMP 2006, Wroclaw, Poland

Paper B – J. Johansson (2007). *A Flexible Design Automation System for Toolsets for the Rotary Draw Bending of Aluminium Tubes*. IDETC 2007, Nevada, USA

Paper C – J Johansson (2008). *Manufacturability Analysis Using Integrated KBE, CAD and FEM*. IDETC 2008, New York, USA

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter provides an introduction to the thesis, defines the contents, and points out the research questions dealt with.

1.1 The importance of design automation for production preparations

Four critical factors for being competitive on the global market have been pointed out: *low cost, short lead-time, improved product performance, and the possibility to adapt products to different costumer specifications* (French, 1999). An automation of the production preparation process will shorten lead-time and make the adaptation of products to different specifications easier. An automation of production preparation would help improving product performance since it enables a common knowledge base, to which quality assurance issues can be targeted. The automation makes it possible to evaluate a big number of solutions.

The ultimate reason for doing research on design automation is to avoid using human engineers from doing routine-like work and instead freeing them to do what they are really good at, creative thinking. Design automation for the production preparation will also help optimise products against production criteria since the production preparation is done in the same step. The design, evaluation and adjustment of toolsets are done automatically to new product specifications. When optimising the product, production criteria are hence available objectives. At the end, the automation of the production preparation process will make companies more competitive on the global market where constant, large economic pressure is the reality today.

1.1.1 Cut lead-time

The fact that shortened product development lead time is one of the main sources of competitive advantage has been realized since the 1980s (Smith, 1988, 1990). It continues to be important today because of global competition, fast changes in technologies and customer preferences, and the increasing development cost of new products. There are great possibilities to cut lead-time in the product development process through the automation of routine work. Although it cannot be said that the everyday work of an engineer is routine, the same data, formulas and rules are applied over and over again to adapt well-established concepts to new specifications. The know-how of this process is the special corporate competence or the special competence of one engineer in the product development team. This is especially true for variant products, a category to which toolsets for production equipments most often can be counted.

1.1.2 Mass-customisation and variant rich products

One of the critical factors for being competitive mentioned above is the possibility to adapt products to different customer specifications. This has led to mass customisation, defined as “producing goods and services to meet individual customer's needs with near mass production efficiency” (Kaplan and Haenlein, 2006). Design automation helps the development of mass-customized products since they have many variants and are a good automation target. Many of the variants can be achieved by configuring components, but many times more advanced tasks need to be completed. Examples include dimensioning using advanced formulas or verifying using simulations. Design automation supports the mass-customisation by automating these tasks.

1.2 Realization of design automation

A design automation system is a computerized system developed to support engineers in the design process. In order to make such systems available, theories from the field of artificial intelligence in computer science are adapted and utilized in the field of product development. Since design automation systems are intended to work within corporations, organizational aspects need to be considered as well.

Automation has a strong potential to cut lead-time for such "routine" work mentioned above. Probably most engineers in that situation have already thought of, and implemented, some kind of automation, commonly by using spreadsheets, databases or algorithmic programs. In all those files, a lot of knowledge is stored. Since this knowledge is applied to the products developed by the company, it will be a part of the brand. Also, the knowledge stored in those files has probably been used for a long time and has thus evolved over time and been proven to work in practical applications. Hence, this knowledge is highly valuable for the corporation and should be captured and secured. Though captured, the knowledge still has to be usable. Further, although secure, it still has to be reachable. The systems built containing the corporate knowledge should be transparent and easy to understand for users and future developers. This means that it is easy for the user to understand not only how the system is built but also the knowledge itself. Thus, there are high requirements for documentation both on the system architecture and the knowledge it contains. In addition to this, Cederfeldt (2007) points out the following requirements that a design automation system should meet: *low effort of developing, low level of investment, user readable and understandable knowledge, transparency, scalability, flexibility, longevity and ease of use*.

1.3 Research project and case of application

The application of rotary draw bending of aluminium profiles has been the target of design automation. This application is beneficial because rotary draw bending is a common forming process in the industry. Adding knowledge on how to automate the design of toolsets for that process is of great importance itself. Further, the rotary draw bending process is representative of other metal forming processes, making them plausible applications for design automation using the same methods.

There are some differences between regular tube bending and aluminium profile bending. Firstly, the components in the toolset are sometimes made from nylon. Secondly, tubes are only a subset of general profiles and, hence, commonly used heuristic rules will probably fail to predict the behaviour of the profile during and after the bend action. Thirdly, the material properties of aluminium are time dependent. The automation of the production process presented in this work is restricted to aluminium profiles with circular sections.

1.4 Research questions

The main research question for the work presented in this thesis has been:

How to apply the concept of design automation to rotary draw bending toolsets for aluminum tubing?

During the project this question has been divided into three sub questions:

1. *How should design automation systems be built to allow a high degree of flexibility?*
2. *How should design automation systems be made able to handle the situation when multiple types of knowledge coexist for a single phenomenon?*
3. *How may design automation systems be built enabling automatic but also flexible finite element analysis of the rotary draw bending of aluminium tubing?*

When answering the research questions, further questions are raised, especially the question of how general the methods are. This question is addressed in the discussion chapter.

In the thesis work, the proposals presented are viewed as hypotheses and described in the chapters along with the test implementations.

1.5 Thesis outline

In Chapter 2, the research method used to conduct the research carried out in this thesis is presented. Chapter 3 gives an introduction to the design automation, which is the research topic, but also to the rotary draw bending, which is the target of the experiments. In Chapter 4, how to build design automation systems with a high degree of flexibility is shown. Chapter 5 presents a method for how to model conflicting knowledge and how to solve situations of conflicts. In Chapter 6, how to automate the finite element analysis process as a local system in a global design automation system is illustrated. Chapter 7 concludes the research work, and a discussion is presented in Chapter 8. Appendix A contains two tables showing the implemented knowledge base. In Appendix B, the knowledge base developed for the design of toolsets for the rotary draw bending in order to conduct the tests is partly presented.

CHAPTER 2

RESEARCH METHOD

In this chapter, the applied research method used during the research project is described.

It has been stated that research conducted on computer systems needs to be empirical (Simon, 1996). This is especially true for large computer systems, since they are too complex to build mathematical theories around without first building and observing them. A method to conduct empirical research is described by Groot (1969) as: Observation – supposition – expectation – testing – evaluation. This method has been adapted in the field of design and in the field of design automation. The adapted process is shown in Figure 1, and is taken from Roozenburg and Eekels (1995). This process starts with a problem definition, followed by observations resulting in a collection of facts. The facts are used to do an induction, resulting in a hypothesis. The hypothesis is use to deduce a prediction. To evaluate the level of truth in the hypothesis, the deduced prediction is tested against the real world. New knowledge is the result of the process.

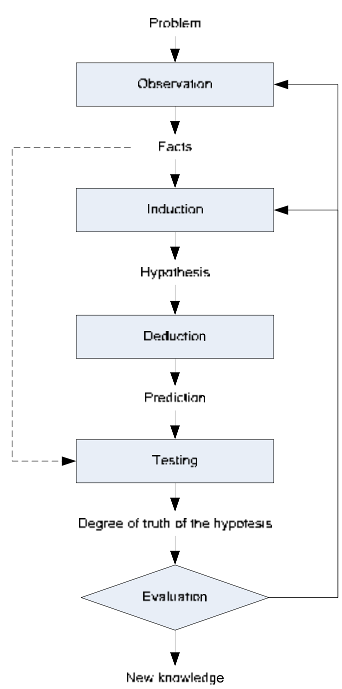


Figure 1: The research method used is the one Roozenburg and Eekels have derived from Groot.

2.1 Applying the research method

Adapting the mentioned research method into the project resulted in the process shown in Figure 2. As seen, the first step is the problem definition, i.e. the selection of research topics. Then engineers are observed to see what activities, rules, and methods are suitable to be automated in the design process. Knowledge about how to create the knowledge base is then induced using the observations of the design process. The induced knowledge, together with background knowledge about design automation systems, is used to plan a design automation system for the selected product development sub-process. That step can be viewed as stating a hypothesis and deducing the system plan.

The subsequent step is to test the system in order to establish some value of truth of the hypothesis. This is not a trivial step. How would a perfect test of a design automation system be designed and executed? One first thought might be to do an implementation and comparing system output to output from the observed product development sub-process. That test would show if the system calculates correctly compared to the original way of doing things. But it does not tell how well the system will perform in a big-scale solution within the organization and how it connects to other processes and systems. Another thought could be to let the engineers give their opinion. However, that often leads to user-interface discussions. To do the ultimate test, two parallel design processes, one traditional and one automated, should be initiated at the same time with the same people and the same environment to see the real performance of the automated system. This is of course not possible. Hence, we have to rely on the comparing outputs test. The test is intended to give insight into how good the system mimics the original process and answers the question of how true the hypothesis is.

Finally, an evaluation of the process of observing and inducing is performed and reflected over to see if there are any mistakes or if anything could be done better. The documentation, evaluations, and reflections are added to the background knowledge of design automation systems and used in the next cycle of empirical research.

The results are validated through close collaboration with industry partners throughout the research work and also through the research community via publications. Verification of a single system is based on experts' acceptance of knowledge put into the system. Over time, a system is also verified through the acceptance of the design proposals generated by the system. In other words, if the engineers using the system seldom change the generated design proposals, it is verified. Otherwise, it is falsified.

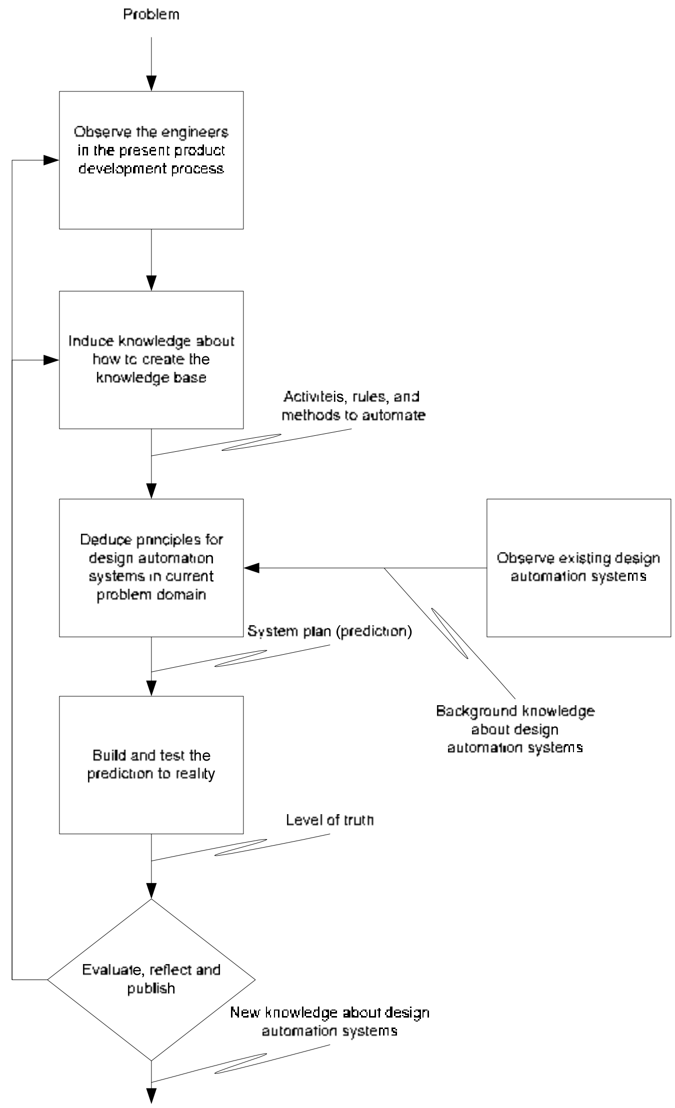


Figure 2: The applied research method was adapted from (Roozenburg and Eekels, 1995).

CHAPTER 3

FRAME OF REFERENCE

The aim of the presented research work has been to apply design automation to rotary draw bending. Hence, there are two supporting fields that need to be reviewed: design automation and draw bending. Design automation is the research topic of this thesis, and the rotary draw bending section is added to give the reader the necessary background information.

3.1 Design automation

Research done in the field of design automation is concerned with questions regarding how to automate the design process by means of computer implementations.

Recent research in design automation has concerned the planning of design automation systems (Cederfeldt, 2007), how to make use of product information in the early phases of the product development process (Boart, 2007), the design process of functional products (Löfstrand, 2007), how to automate producibility aspects from a costing perspective (Elgh, 2007), and design for manufacturing (Sandberg, 2007). A design automation system affects many individuals in a product development company and organizational aspects need to be considered. Research concerning that problem is found in (Catic and Malmqvist, 2007), where knowledge based engineering is intended to be integrated into the product lifecycle management system.

3.1.1 Artificial intelligence and Knowledge Based Systems

Artificial intelligence can be defined as that branch of computer science concerned with the automation of intelligent behaviour. Creative thinking is not the scope of design automation. Rather, design automation involves the automation of routine work in order to free humans to do creative thinking. Anyhow, some of the methods found in artificial intelligence have proven to be successful when applying them to different routine-like tasks. One example is the design of kitchen layouts. The same types of components are used in every kitchen, and the layout is controlled by some few rules. But creating a computer implementation designing good kitchen layouts is a hard task using traditional procedural programming. To solve the problem, several solutions are generated according to implemented rules and searched using a search method. Another problem well suited for methods from the artificial intelligence is the diagnosis of systems through logical reasoning. For instance, in automating the diagnosis of why a car does not start, it is possible to define a rule base and search that for possible failures. In the two examples, knowledge is stored in a knowledge base searched by an inference engine; the complete system is called a knowledge-based system. The knowledge-based system is a sub-category of the more general category intelligent systems (see Figure 3) that is a result of the field of artificial intelligence. The knowledge-based systems can further be divided into sub sections including agent-based systems, expert systems, and fuzzy logic systems.

More about artificial intelligence in general and in practice is found in (Fulcher, 2006; Jones, 2003; Luger, 2005).

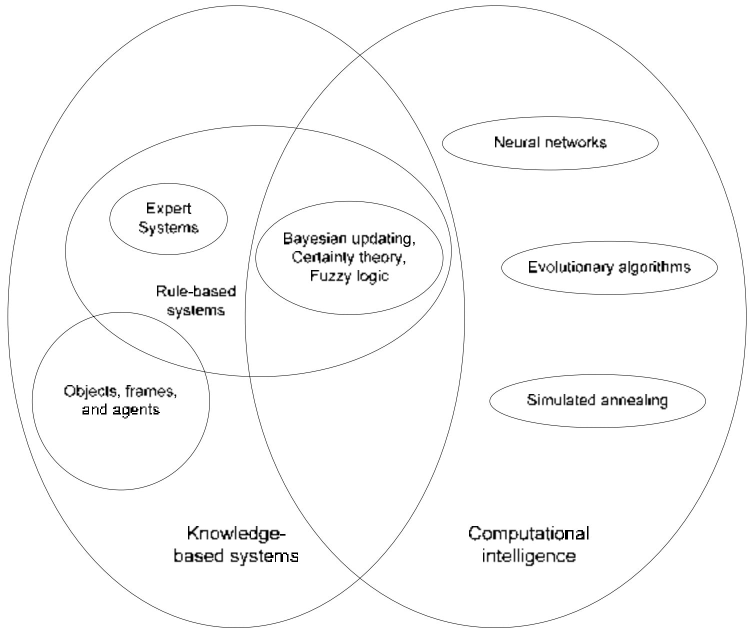


Figure 3: In (Hopgood, 2001), a categorization of the intelligent system software is done. The knowledge base systems category is a big share of the intelligent system domain and is broken down in several sub-domains.

3.1.2 Knowledge Based Engineering

Knowledge based engineering aims to automate engineering tasks by means of knowledge based systems. The fact that the concept of KBE has many definitions might be due to the wide area of the knowledge based systems and their many sub-categories. The definition of KBE adopted in this thesis work is the one stated by Stokes (Stokes and MOKA Consortium, 2001): *"The use of advanced software techniques to capture and re-use product and process knowledge in an integrated way."*

A general structure of a knowledge based system is shown in Figure 4, adapted from (Hopgood, 2001). As seen in Figure 4, the two keystones in a knowledge based engineering system are the knowledge base and the inference engine. The knowledge base is comprised of facilities to store knowledge in the sense of information in context. This means that structuralized data is stored with its context in a way that makes it possible for the inference engine to make use of it. Hence, the knowledge is separated from the routines (the inference engine) that make use of the knowledge.

It is possible to define the knowledge base in different ways, different knowledge representations. The knowledge base must be machine-readable. This means that the knowledge base is designed to make the system able to automatically reason based on the knowledge. It would of course be highly beneficial if a machine-readable knowledgebase also was human-readable, which is far from common. In this thesis work, two knowledge bases are implemented, one containing rules and the other containing knowledge objects.

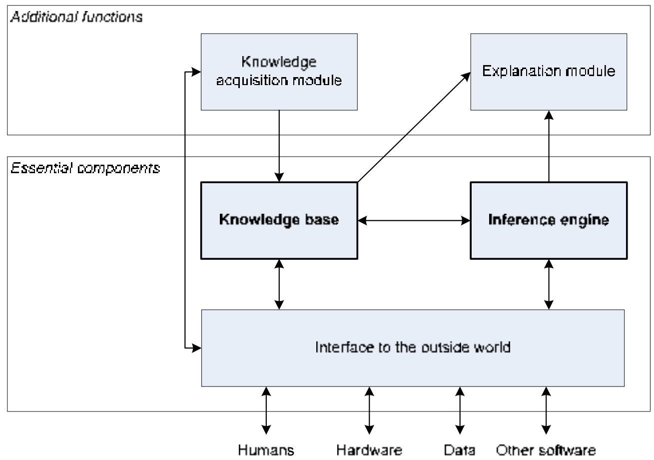


Figure 4: The main components in a knowledge-based system are the knowledge base and the inference engine. Techniques for how to build the inference engine and how to represent the knowledge in the knowledge base are adapted from the field of artificial intelligence (Hopgood, 2001).

Furthermore, it is possible to search the knowledge base in different ways, search methods. In this thesis work, the forward chaining (kitchen example mentioned in the previous section) and the backward chaining (diagnosis example mentioned in the previous section) search methods are interesting and described in detail in Chapter 4.1.2. Further information on knowledge representation and inference engines in the scope of engineering is found in (Hopgood, 2001; Sriram, 1997).

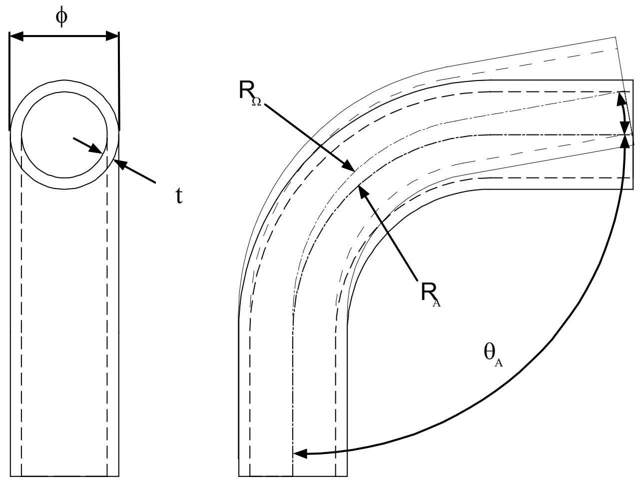
The following terms are used in this thesis in connection with KBE-systems: a *triggered* object is an object with all input parameters known, but with at least one output parameter unknown; a *solved* object is an object with all output parameters known; the *conflict set* is the collection of all triggered objects; and, to *fire a rule* is to execute, or evaluate a rule.

3.1.3 Integrating knowledge for analysis applications

Knowledge based engineering is about integrating knowledge. One specific type of integration interesting in the scope of this thesis is the finite element analysis integration. Some examples of knowledge based engineering systems exist where the finite element analysis process has been integrated. Sellgren (1999) developed a framework for simulation-driven design, in which simulation models were extracted based on the CAD-model relationships. Chapman and Pinfold described how to use KBE and FEA for the design automation of a car body (Chapman and Pinfold, 2001), and a system was presented by Hernández and Arjona, (2007) that automatically designs distribution transformers and that uses FEM automatically. The design process of different jet engine components has also been the subject for design automation using KBE (or KEE) integrated with FEA (Boart, 2007; Sandberg, 2007). The benefit of integrating the FEA to KBE-systems is that the natural synthesis and analysis process is captured (see further Chapter 6).

3.2 Draw bending – a short introduction

Since rotary draw bending is the application of the research project presented in this thesis, a short introduction to that subject is merited and given in this section. The nomenclature shown in Figure 5 is used.



Symbol	Description	Symbol	Description
σ_y	*Material yield strength*	L	*Feed preparation length*
ϕ	*Tube outside diameter*	t	*Original wall thickness*
r	*Tube mean radius*	t_{max}	*Maximal wall thickness after bending.*
R_A	*Nominal centre line radius*	t_{min}	*Minimal wall thickness after bending.*
R_Ω	*Centre line radius after spring back*	$k=R/\phi$	*Bending factor*
θ_A	*Nominal bending angle*	β	*Transition zones (see figure 31)*
θ_Δ	*Angular spring back*	δ_c	*Average shrinkage rate*
θ_Ω	*Final bending angle*	δ_{co}	*Shrinkage rate for outer semi-circle*
E	*Young's modulus*	δ_{ci}	*Shrinkage rate for inner semi-circle*
M	*Necessary moment for making the bend*	C	*Average perimeter length*

Figure 5: Nomenclature for design variables.

A rotary draw bending tool is usually built from five different main parts (components for applying boost pressure will not be addressed in this work). Three of these parts, form die, clamp die, and pressure or follower die, are necessary for making a bend. The two other parts, mandrel and wiper, are used when there is a risk of wrinkling and/or flattening. Some of those components come in different variants. In figure 6, a tool set-up is shown where a form die with insert and a flexible mandrel with regular pitch are used.

When designing a toolset for rotary draw bending, a few typical problems arise. Circumferential decrease means that the tube diameter has a tendency to decrease. Wall thickness distortion means that the wall thickness on the inner side of the bend will be increased and decreased on the outer side. The wall thickness distortion displaces the section neutral axis affecting the feed preparation. When releasing the clamp after bending, elastic deformations in the material make the tube spring back both in radial and in angular direction.

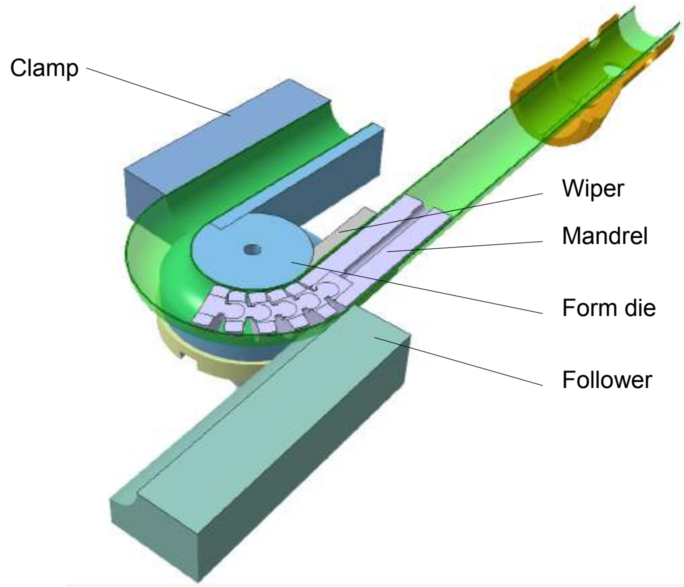


Figure 6: A tool set-up for tube bending can contain up to five main components that exist in many variants.

The mentioned phenomena have been studied to varying extents, and a summary of studied literature is found in Table 1. When searching the literature (handbooks and scientific publications), it is clear that duplicates of rules for some of the phenomena exist. In Appendix B, the spring back and the wall thickness distortion are described in detail addressing heuristic, analytical, numerical and experimental rules and data for these two phenomena.

Phenomenon	*Heuristic rules*	*Analytical rules*
Angular spring back	(Cone, 2007)	-
Circumferential decrease	-	(Tang, 2000)
Developed length	(Cone, 2007; Gillanders, 1994; Miller, 2003)	(Tang, 2000)
Radial spring back	-	(Abdel-Malek and Maropis, 1998; Gillanders, 1994; Miller, 2003)
Wall thickness distortion	(Bradley, 2007; Cone, 2007; Gillanders, 1994; Miller, 2003)	(Tang, 2000)

Table 1: Several works have been reviewed to find rules for designing rotary draw bending toolsets. Sometimes there exist multiple rules for a phenomenon.

3.3 Existing systems for the automated design of draw bending toolsets

When it comes to the automation of the design of rotary draw bending toolsets, three different works have been found. They are briefly described and commented upon below.

3.3.1 A rule-based system

Abdel-Malek and Maropis (1998) performed the earliest of the research works found. In that work, theories from seamless design to manufacture (SDTM) were used to build a system for manufacturability analysis. That work presents an automated system for design-to-manufacture which can perform post-fabrication operations concerning bending and other processes. The system is a rule-based system integrated with a CAD-system (see Figure 7). The system enables the automatic design of the toolset assembly and generation of blue prints and NC code for appropriate mechanical parts. To make it possible to create and edit rules, a user interface was developed using three different types of “nodes” for displaying/retrieving data to/from the user, for performing calculations, and for branching to other “nodes”. The user interface was developed using the development tool Symbologic Adept, the rule base was developed using AutoLisp, and routines for the generation of drawings and NC-code were developed using the C language.

Programmers skilled in AutoLisp and the C language can change the knowledge base in the described system, having competence in knowledge-based systems. Since the knowledge base is hard for humans to interpret, documentation is essential. The same is true for the in-house developed routines.

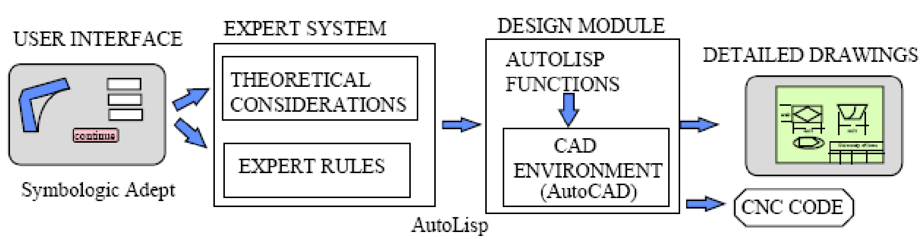


Figure 7: An overview of the system described in (Abdel-Malek and Maropis, 1998), from which the picture is taken.

3.3.2 A goal-driven system

Jin et al. (2001) performed the second work found describing the design automation of toolset for the rotary draw bending. It describes an object-oriented approach combined with an inference engine using a goal driven search mechanism. The objects created to build the knowledge base were collections of rules as they appear in a rule-based system. Semantic networks were used to describe the objects internal rule-set. The development language LEVEL5 OBJECTS was used to build the system. Figure 8 shows an example of the internal rule-set of an object for determining whether the final wall thickness will meet product requirements. Figure 9 shows how the inference mechanism in LEBEL5 OBJECTS connects different objects.

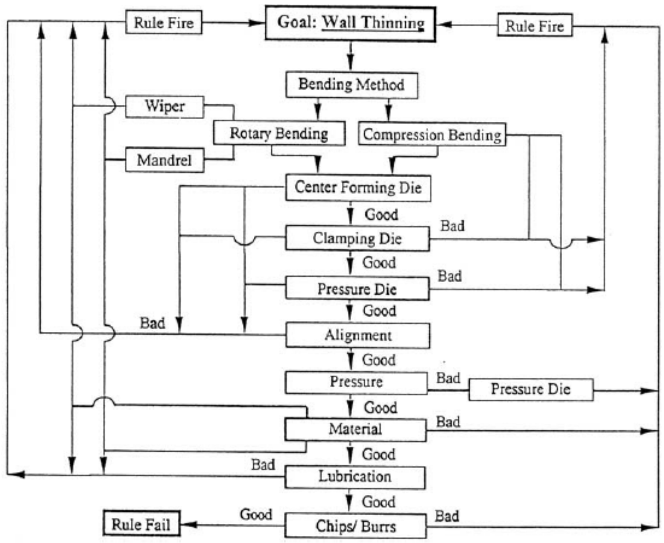


Figure 8: An object consisting of a set of rules described by a semantic net. The system is described in (Jin et al., 2001), from which the picture is taken.

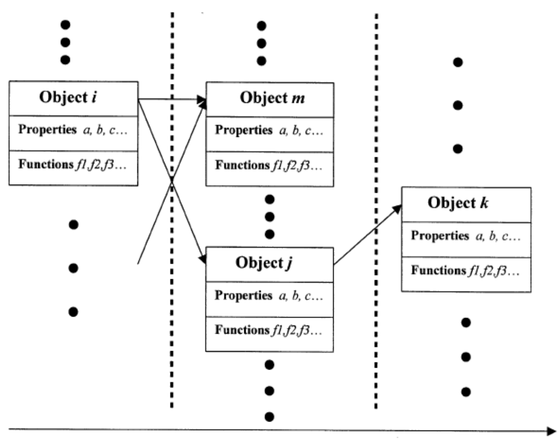


Figure 9: The system developed by Jin et al was implemented using object oriented programming to build the knowledge base and user interfaces. The picture is taken from (Jin et al., 2001).

Since the declarative language LEVEL5 OBJECT was used to implement the knowledge, changes in the knowledge base can be done straight forward by a programmer skilled in that language. Since the objects can be cumbersome to read even to skilled knowledge engineers*, documentation is essential.

3.3.3 A fuzzy logic system

The third research work found describing a system for the automation of toolset design for the rotary draw bending was done by Strano (2005). That work presents a system using fuzzy logic to select correct tool components. The system is executed in two steps, where two different knowledge bases are used. In the first step, calculations are done to

* An engineer who builds and maintains KBE-systems is often referred to as a knowledge engineer.

get values on variables to feed the knowledge base used in the second step. The second step is a fuzzy logic system. The approach when building the knowledge base for the fuzzy logic system was to transform all acquired knowledge into decision tables and in which selections are done using production rules (if … the … else). Figure 10 shows a section of the rule-block used to select appropriate type of mandrel. Figure 11 shows the structure of the fuzzy logic system. Further, a module for self-learning was added to make the system able to improve its selections.

#	IF					THEN	
	DoB	RISKflat	RISKthin	RISKwrink	WF	DoS	Tm
43	max				term5	1.00	M
44	max				term6	1.00	M
45	max				term7	1.00	M
46	max				term8	1.00	M
47				medium		0.60	M
48				large		0.90	M
49		large				0.33	M
50		very_large				0.33	M
51	min				term6	0.70	TW
52	min				term7	0.80	TW
53	min				term8	0.90	TW
54	min				term9	1.00	TW
55	verylow				term6	0.80	TW
56	verylow				term7	0.90	TW
57	verylow				term8	1.00	TW

Figure 10: A part of a rule-block implemented in the system developed by (Strano, 2005), from which the picture is taken.

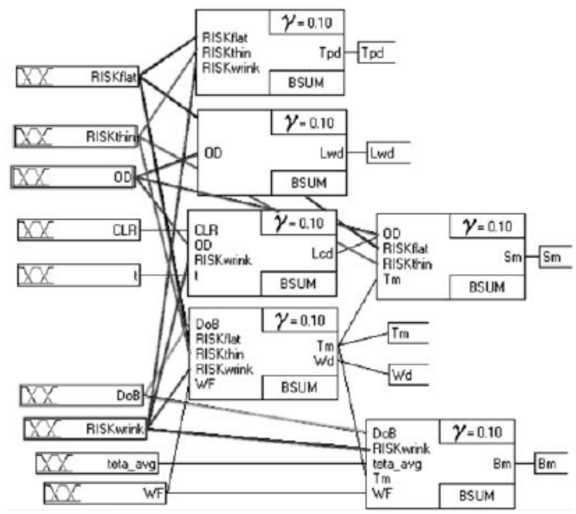


Figure 11: An overview of the fuzzy logic system described by (Strano, 2005) , from which the picture is taken.

The presented system is hard to maintain since predicting what a change in the knowledge base would do is not a straightforward task. This is due to two facts. First, it is hard to make global changes when decision tables are used the way they are in the first knowledge base. Second, the knowledge stored in the fuzzy logic structure (that is actually doing the design proposal) and in the self-learning module is far from human-readable. Making changes in these knowledge bases is very complicated.

3.4 dentified knowledge gaps

It is clear that there are some knowledge gaps when reviewing the existing systems for the automation of the production preparation for rotary draw bending. The systems reviewed are hard to maintain due to their complex knowledge bases or due to the knowledge being transformed into something humans are not able to read. The systems are inflexible when adding or changing the knowledge because in such cases major parts of the system need to be changed. None of the systems are able to verify the generated design proposals.

Additionally, it is found that multiple rules can be used for a single problem when reviewing the knowledge used by engineers in the observed development process. Further, the design proposals in the manual process are most often analyzed using the finite element method.

This implies the development of methods for building design automation systems that have a high degree of flexibility, the ability to add redundant knowledge to the knowledge base, and the ability to analyze their own design proposals.

CHAPTER 4

DESIGN AUTOMATION SYSTEMS WITH A HIGH DEGREE OF FLEXIBILITY

Since it is of paramount importance for a design automation system to have a big degree of flexibility, this has been one of the research questions and is addressed in this chapter. Two ways of achieving flexibility are shown. One method is by applying the object orientation approach to the knowledge base. Another is by applying the object orientation to the product structure. Finally, guidelines are stated telling when to implement the knowledge using a CAD-integrated KBE-system and when to use a stand-alone KBE- system.

4.1 A modular structure of the knowledge base

To make a system truly flexible, all the parts building that system must be autonomous. Hence, to make the knowledge base truly flexible all the chunks of knowledge must also be autonomous. This implies the use of object orientation together with an inference engine.

4.1.1 Knowledge objects

Object-oriented programming offers the possibility to develop highly flexible programs, and a class of objects called knowledge objects* is proposed (see Figure 12). The least amount of information a knowledge object contains is a list of input parameters, a list of output parameters, and a method for process input parameters to output parameters. Other fields may be added to a knowledge object. Proposed additional fields are *constraints*, *owner*, *categories*, *precision,* and *comments*. *Owner* is used to trace who are responsible for the knowledge object and its method (the task it performs). The field *categories* can be used to sort knowledge objects into groups. *Comments* are used to add information usable for explanation extractions and debugging facilities. Finally, the list of *constraints* and the *precision* value is used to allow knowledge bases to contain conflicting knowledge objects (see further in Chapter 5.2.1).

When implementing the knowledge objects, they should be defined in a way that makes them autonomous. Since the methods used to process information preferably are external software applications, the applications should be selected keeping in mind the list of requirements imposed on the design automation system: *low effort of developing, user readable and understandable knowledge, longevity, and ease of use* (Cederfeldt, 2007). The benefit of developing knowledge objects that are autonomous using common wide

* Objects are closely related to frames. The difference is that a frame is passive in the sense that it doesn't perform any tasks itself. An object, on the other hand, stores information about itself and performs certain tasks (Hopgood, 2001).

spread applications as methods is that the knowledge can be used manually without the design automation system, and there will always be people skilled enough to use the very same knowledge the design automation system does. It makes the knowledge more human-readable (see section 3.1.2).

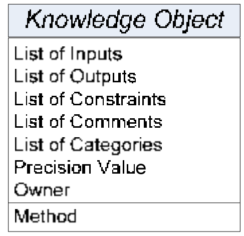


Figure 12: In the proposed method, the keystone to achieving a higher degree of flexibility in design automation systems is a class of objects called knowledge objects, with two lists of parameters and a method. A knowledge object may also have other fields, such as: constraints, comments, categories, precision value, and owner.

The anatomy of the knowledge used when designing toolsets for the rotary draw bending implies the use of knowledge objects. Several distinct tasks are identified and implemented as knowledge objects.

4.1.2 Inference engine

An inference engine is needed in order to use the knowledge stored in the knowledge base. The engine is used to arrange the knowledge in the knowledge base in an executable order. Two main types of search based inference engines exist: forward and backward-chaining (see Figure 13). A forward-chaining (also called data-driven) mechanism uses the information initially presented to fire all rules it can ever find. The method has two steps. In the first step, triggered rules are listed. In the second step, an appropriate rule from the triggered ones are selected and fired. After firing the selected rule, all triggered rules are listed again and so on, until no triggered rules are found. If knowledge objects are used to build the knowledge base, the inference engine searches for knowledge objects with all input parameters known and selects one of the found knowledge objects to execute the method defined in that knowledge object to calculate output parameters using the input parameters. When the method has run, the stock of known parameters is updated and a new search for executable knowledge objects is begun (see Figure 14).

A backward-chaining mechanism (also called goal-driven) is fed with information of a final state. The mechanism then searches backward to see how to end up at that state. When knowledge objects are used, this means searching the parameter dependencies to see what information is needed to retrieve wanted output parameters.

When using knowledge objects, the backward-chaining mechanism makes the system more effective. This is because executions of unnecessary methods are avoided. All the design parameters need to be calculated when designing toolsets for the rotary draw bending, implying the use of a forward-chaining mechanism.

A key issue with inference engines is how to make the selection of an appropriate rule (or knowledge object) out of the found triggered ones. Different approaches exist: depth first, breadth first, and combinations of these two using heuristics.

In order to execute the knowledge objects, a connection layer is added to the inference engine. API-dependent routines are added in the connection layer making it possible to send and retrieve information from the external applications.

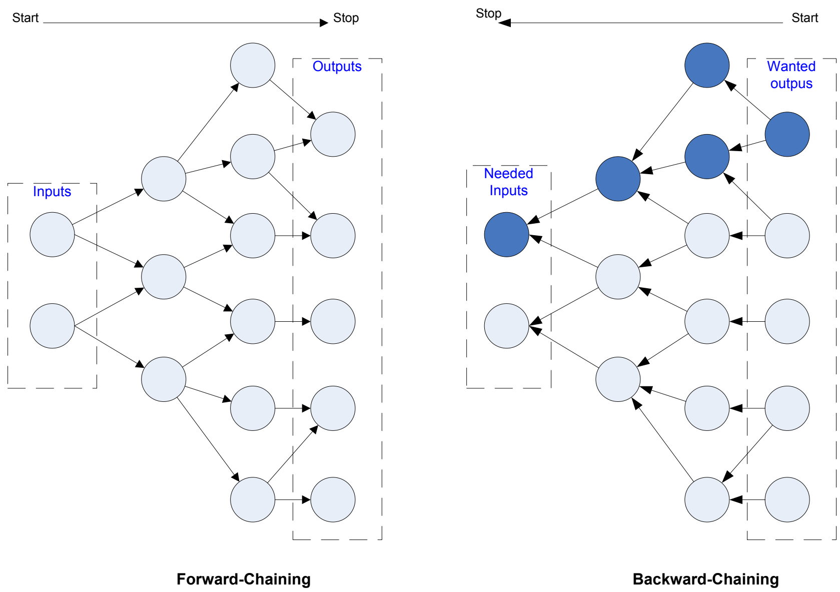


Figure 13: Two main types of search-based inference engines exist: forward and backward-chaining. The first one uses the inputs to retrieve all possible data. The second one searches the knowledge base to see how to end up at a state presented to the system.

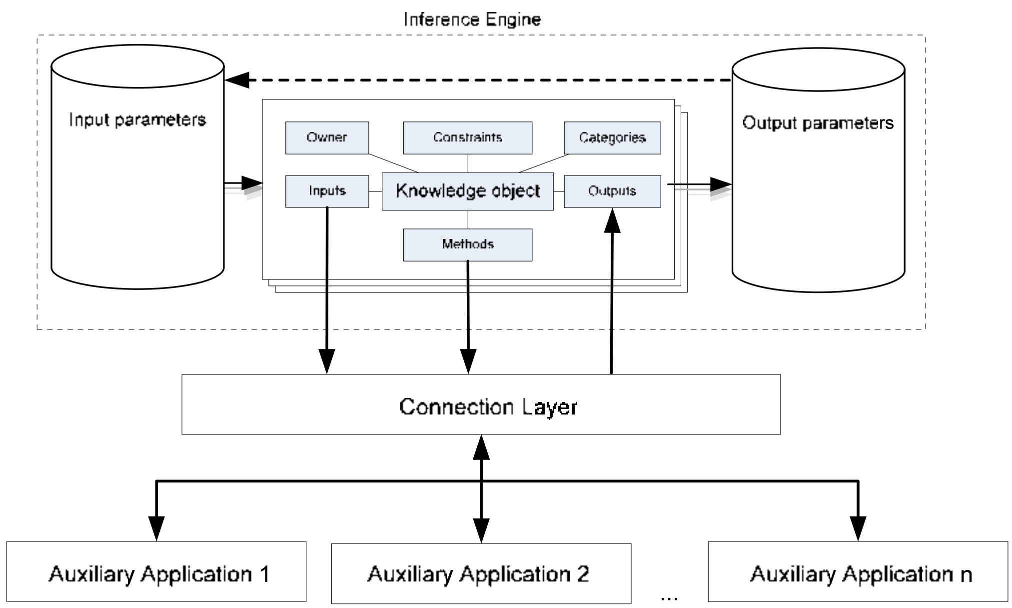


Figure 14: The collection of knowledge objects is solved using an inference engine that has a connection layer added where API-dependent functions are added in order to connect the external applications.

4.1.3 An event driven inference engine

Event handling is available in today's operating systems. And here it is proposed that the inference engine should make use of these advanced functions in the operative systems. That gives an event-based forward chaining search mechanism working as follows. When a parameter is changed, an event is raised in the system notifying that a change has occurred. This triggers an update of the conflict set. If there still are objects left in the conflict set, an object is selected according to implemented rules for selection in order to execute the object. When the object is executed, its output parameters are changed and the conflict set is updated, and so on (see Figure 15). The benefits of using events are the following: when implementing the inference engine, a lot of loop algorithms are avoided and when running the system the inference engine is triggered automatically on change, no extra button clicks are needed (unless functionalities for avoiding automatic updating are implemented and activated in the system).

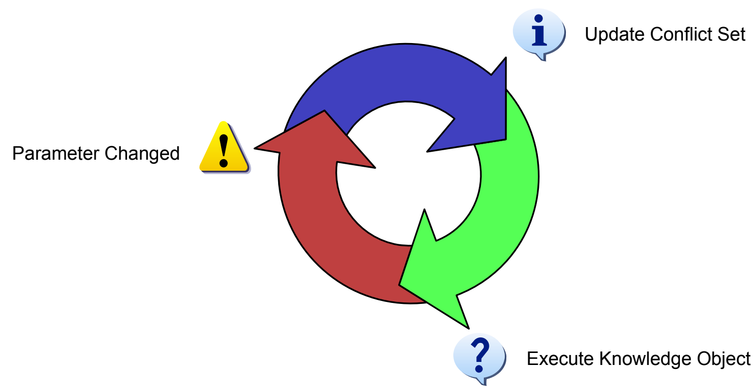


Figure 15: Using event handling when implementing an inference engine means that routines are run on changes. The change of a parameter triggers the system to update the conflict set. If it is not empty, a knowledge object will be fired causing parameters to change and so on. The loop is driven by parameter changes.

4.1.4 Global and local automation

To allow design automation systems to have a high degree of flexibility, the proposed method in this section is a modular knowledge representation interpreted by an inference engine. The modules themselves might be local design automation systems though. A knowledge module could, for instance, contain an algorithm for finding the shortest path using simulated annealing, and another module could connect to a configuration system getting a product structure. The global design automation system contains local design automation systems. It can be a semi-automatic system containing local isles of fully automated design automation systems connected by graphical user interfaces. The global system can also be a fully automated system containing isles of fully automated design automations systems that are fully integrated. This means that different levels of automation can be achieved and considerations have to be done to what extent and in what steps the design process should be automated and to what cost.

4.1.5 Evaluating the flexibility when using knowledge objects

To evaluate the flexibility when using knowledge objects, an implementation was done. The implementation uses the class knowledge objects, as described in this chapter, and an event driven forward chaining inference engine as well. The production preparation for rotary draw bending was the application. Since additional features were added in connection with the other research questions, the implementation is described in Chapter 5.3.

4.2 Adapting object orientation to the product structure

The other way to achieve a higher degree of flexibility in the design automation system is to adapt the object orientation approach on the product structure. Doing so is especially beneficial when applying the concept of design automation systems to production toolsets. When it comes to modelling production tools using CAD integrated KBE-systems, it is hard to decide where to put the knowledge and the parameters/variables. To illustrate this, we can take one example. It is a toolset for rotary draw bending where some parameter might be needed in several or all of the components. In Figure 16, all five bending tool components are shown with associated parameters. It is clear that some parameters appear in several or all of the components. This indicates that there exist coupled relationships. In other words, changing the outer tube diameter will affect all components and trigger a sequence of dependent rules and calculations. An object-orientated view of the product is proposed to solve some issues. All components are viewed as objects with attributes (input parameters in this case). This is easily achieved in any parametric CAD-system.

When using the object-oriented view of the components, the parameters are put on an appropriate level in the product structure. In the case of rotary draw bending toolsets, the parameters represented in several components are put at the top level in the product structure. All rules and calculations concerning these top-level parameters are also put at the top of the structure. When changing parameters at top-level, they are passed down to sub-level components, either using connections in the CAD-system or using macro programming (see further Chapter 3.2 in Appended Paper 1). Again, the key idea is to make all the parts in the system autonomous. Making all the components autonomous (in this case letting each component contain all necessary parameters for that component) will make the geometric product model more robust. It will then be easy to replace components without major failure.

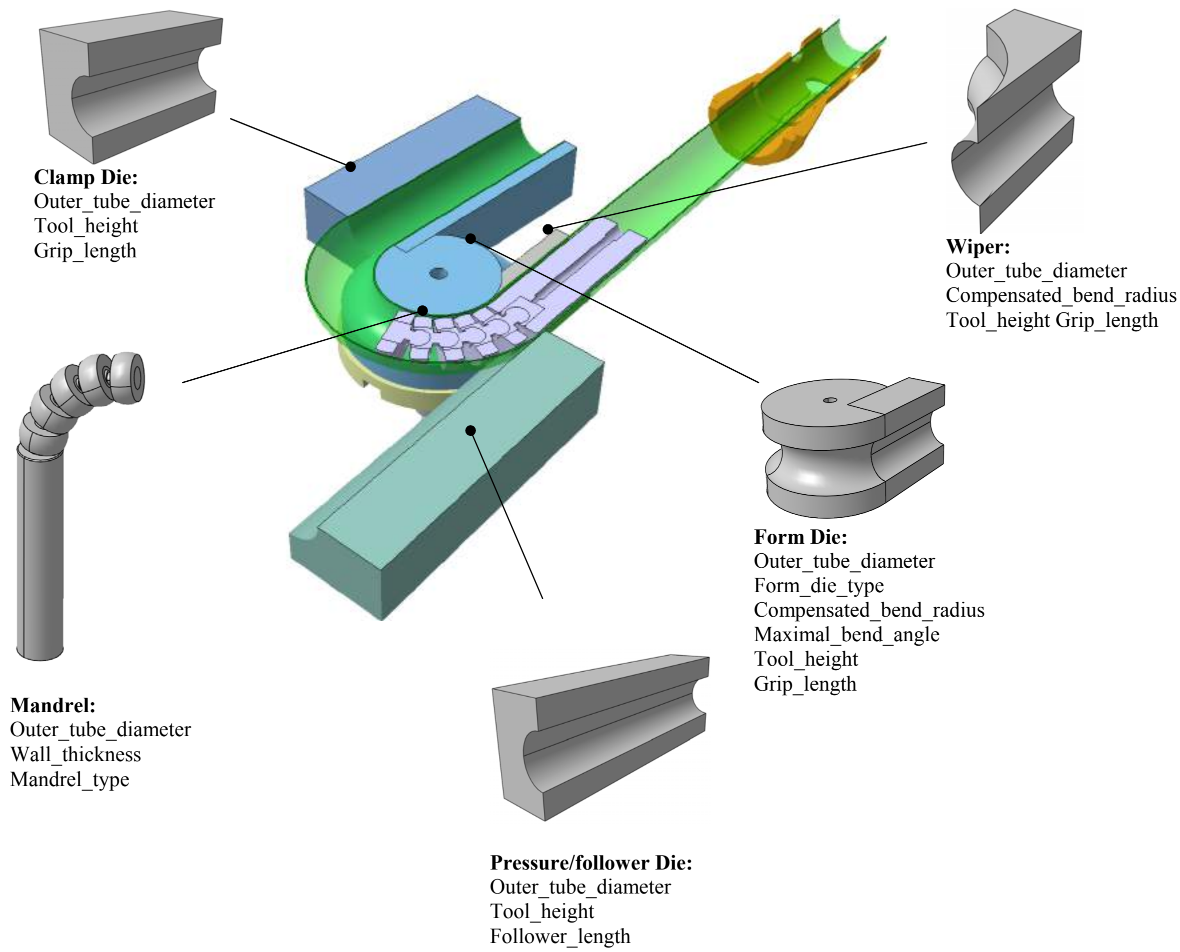


Figure 16: All parameters that a single component is associated to are viewed as attributes of objects. Since several parameters can be related to several components, parameter instances are made from the top-level of the product structure on which governing rules are applied.

4.2.1 Evaluating the flexibility when adapting object orientation on the product structure

In order to try the object-oriented approach on the product structure in a CAD-integrated KBE-system, a generic geometric model of a toolset for the rotary draw bending was created in CATIA and knowledge was added in CATIA/Knowledgeware.

The main idea when building a design automation system for rotary draw bending in CATIA/Knowledgeware was to use the internal programming language to build methods executing external software applications to calculate different design variables (similar to the approach described in Chapter 4.1.1). The system contained methods for executing spread sheets to select correct types of components, material data, and bending machines. It also included methods for executing algorithmic programs to calculate developed length, minimal wall thickness after bending, angular spring back, and section modulus. Some factors were calculated using the internal functionalities in CATIA/Knowledgeware.

The knowledge implemented in the CATIA\Knowledgeware system was heuristic rules and rules analytically derived from fundamental physical laws as found in the literature. More details are found in Appended Paper 1. The knowledge base used in the applications is presented in the Appendix B. Two examples of output are shown in Figure 17.

Parameter	*Example 1*		*Example 2*	
Outer_tube_diameter	45 mm		60 mm	
Tube_wall_thickness	5 mm		2 mm	
Wanted_bending_angle	180°		180°	
Section_modulus	4.74 cm^3		3.965 cm^3	
Tube_material	Sapa 6063-T4		Sapa 6063-T4	
Wanted_bendradius	80 mm		80 mm	
Results	*Heuristic*	*Analytic*	*Heuristic*	*Analytic*
Mandrell_type	Plug	N/A*	Regular Pitch	N/A*
Bending_moment	434 Nm	618Nm	363 Nm	584Nm
Minimal_wall_thickness	3.9 mm	4.3mm	1.5 mm	1.6mm
Developed_length	251mm	239mm	251mm	228mm

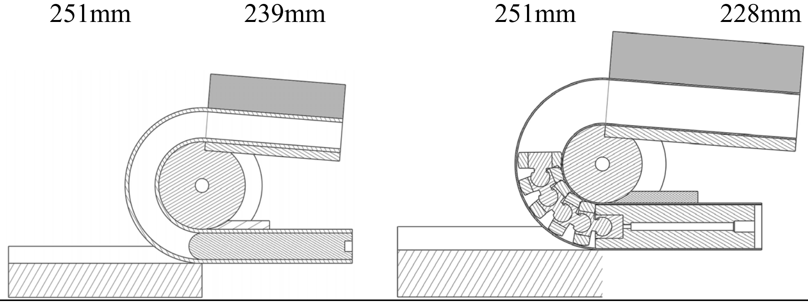


Figure 17: Two examples of output from the system. Total time for set-up is a few minutes, where most of the time is spent typing in parameter values. In the heuristic rules, different factors are used when mandrel types are changed (1.3 for plug and 1.6 for mandrel). (*Was not implemented)

4.3 Stand-alone and CAD-integrated KBE-systems

It is possible to implement the knowledge base in stand-alone KBE-systems or into CAD-integrated KBE-systems. In Figure 18, an implementation using a CAD-integrated KBE-system is shown. Figure 19 shows a KBE-system that is stand-alone and that connects to a CAD-system when a geometric representation is needed.

When using a CAD-integrated KBE-system to implement the knowledge base, the rules will be listed in the model-tree among the different features. This can be valuable since it is easy to see what geometries the rules are connected to and the user will feel familiar with the user interface. But the knowledge base in such a system will be cumbersome to understand when the knowledge base contains a vast number of rules compared to the number of geometry features. This is especially true if many of the rules do not deal with geometry. If that is the case, a stand-alone KBE-system should be used.

Another issue to consider is that when using a CAD-integrated KBE-system, the knowledge is bound to the CAD-system and the knowledge base will be hard to translate to other CAD-systems. In the stand-alone KBE-systems, the knowledge is managed and design proposals can be generated in native or neutral CAD-formats. It can be hard to implement a knowledge base containing mostly geometric relationships into a stand-alone KBE-system, though. In Figure 20, an overview of when to use a CAD-integrated or a stand-alone KBE-system is presented.

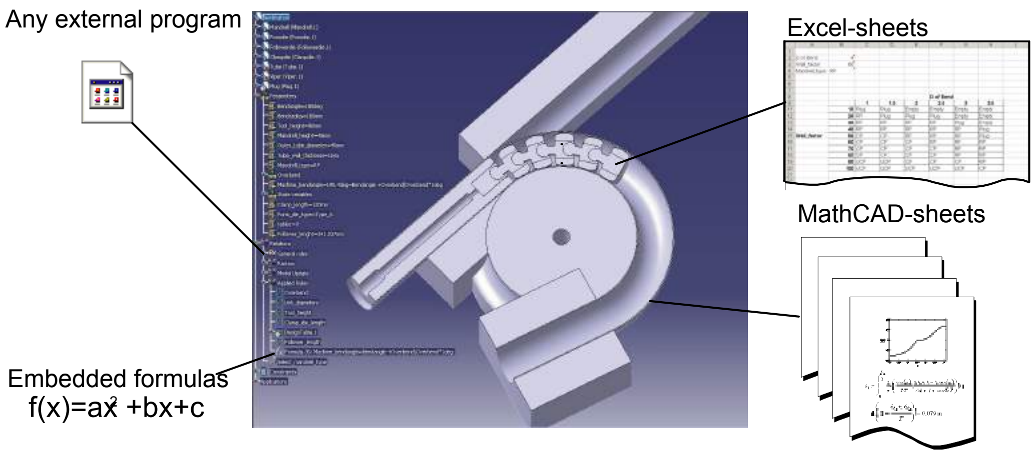


Figure 18: To evaluate the flexibility when using a CAD-integrated KBE-system, CATIA was used to create a generic model of a toolset for the rotary draw bending. Knowledge was added to that model using CATIA\Knowledgeware.

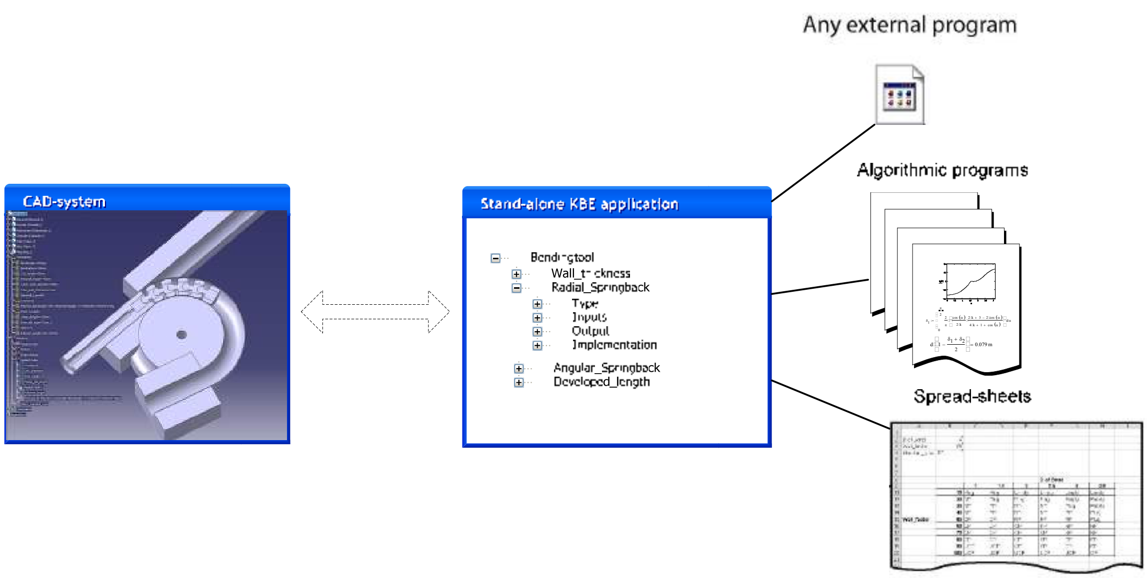


Figure 19: To evaluate the flexibility when a stand-alone KBE-system is employed, an inference engine was developed allowing the use of knowledge objects. The design proposals were communicated to the CAD-system using the API and CAD-templates.

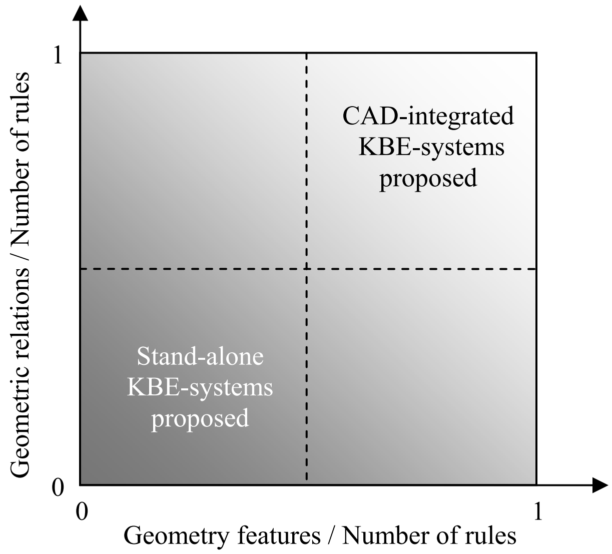


Figure 20: A stand-alone KBE-system is proposed when the number of geometrical relationships compared to the total number of rules in the knowledge base is low and when the number of geometry features compared to the total number of rules is low. Otherwise, a CAD-integrated system is suitable for the implementation of the knowledge base.

CHAPTER 5

HANDLING THE SITUATION WHEN MULTIPLE TYPES OF KNOWLEDGE COEXIST

Often there exist several sources of knowledge, and sometimes there are interferences between these sources. Developing methods for handling such situations will make the design automation systems more adaptable to different situations. In this chapter, the author shows how such interferences in a knowledge domain might occur and presents a proposal for how to solve these situations. Finally, the proposal is tested by implementation.

5.1 Different sources of knowledge and meta-knowledge

It is possible to calculate a single variable in different ways. Sometimes a heuristic rule can be used, or rules analytically derived from fundamentals. But it is also possible to do FEM-calculations or experiments to evaluate an interesting design variable. In addition to these four types of knowledge, an engineer also needs to have the capability to decide when to use what knowledge; this is called meta-knowledge or knowledge about knowledge.

When more than one type of knowledge source is available in a knowledge base, the question of when to use what rule arises. In one state, the system may be executed in order to do a quotation calculation with only a small set of input parameters available. In the next step, detailed design is the purpose of running the system, with high accuracy as the main focus and with a larger set of input parameters available. Different kinds of knowledge are used in those different contexts, and implementing meta-knowledge would allow for flexible use of the system. A description of the different types of knowledge is presented below.

5.1.1 Rules of thumb, heuristics

Heuristics is generally found in different handbooks or company standards and is based on skilled engineers' experiences. Usually they are easy-to-use relationships that are valid for a small range in the design space. The heuristic rules seldom explain why things happen. The benefit of heuristics is that they have often proven to be correct enough and that they enable fast computer implementations. In reality, many design processes and design automation systems are built on this kind of knowledge (see Chapter 3.3).

5.1.2 Knowledge analytically derived from fundamental laws of physics

Rules analytically derived from fundamental laws of physics tend to be more complex than heuristics. However, they have the benefit of explaining why things happen and are more general. One benefit of analytical rules is that they are fast to execute when finally implemented in a computer system. Even though analytical rules are built on fundamental physical laws, they still are idealizations of complex problems.

5.1.3 Finite element analysis

There exist a number of numerical methods for solving different engineering problems. Probably the most common method is the finite element method (FEM). FEM can be used to solve many engineering problems. However, the results depend greatly on what mesh density and elements are used, what boundary conditions are prescribed, what material model is used, and what time step is set.

Using FEM appropriately will give highly reliable data. The drawback is that, compared to heuristic and analytical knowledge, FEM is costly to use both in money, competence, and time. In addition, it does not explicitly answer the question of why things happen. The benefit is that FEM allows full control over the process so it is easy to scroll in time and space, doing sections and plotting different parameters.

Worth mentioning in this context is that simulation-tools are comparable to instruments for measurements, i.e. they must be calibrated. This calibration is done via result feedback. Over time an accurate model will be developed.

5.1.4 Experimental data

Reality is the truth. Thus, making trial manufacturing will give reliable data. The drawback is that experiments are expensive, have a limited range and do not answer the question of why things happen. To make empirical data usable, experiment planning has to be done beforehand to isolate interesting parameters.

5.2 Enabling knowledge adaptability

In this section, a proposal is made on how to solve the situation when multiple types of knowledge coexist for a single phenomenon. The proposal is based on the extension of the knowledge objects and the usage of a common parameter list within the system.

5.2.1 Extending the knowledge objects

In Section 4.1.1, knowledge objects were introduced to represent the knowledge in the knowledge base. A list of constraints and a precision value needs to be added to the knowledge object class in order to make it possible to have multiple knowledge objects pointing to the same parameter in the knowledge base. The constraints tell when the knowledge object is applicable, and the precision value tells how good the knowledge outputs are. The precision value ranges between 0 and 1 and might be changed in the design space.

5.2.2 A common parameter list

Knowledge objects use external applications as methods. To pass information to the external applications, meta-data* is needed. That meta-data is stored in local instance

* Do not confuse with meta-knowledge.

parameters within the knowledge objects. The calculated parameter values are stored in a global list of parameters. For example, in Figure 21, two different knowledge objects calculate a value for the parameter I as an output. Knowledge object number 1 uses application number 1 and needs some information on where to put the inputs and how to retrieve the outputs. Equally, knowledge object number 2 needs information on how to send and retrieve. That meta-data for the parameter is stored in the parameter instance in the knowledge object, while the parameter value is stored in the common parameter list.

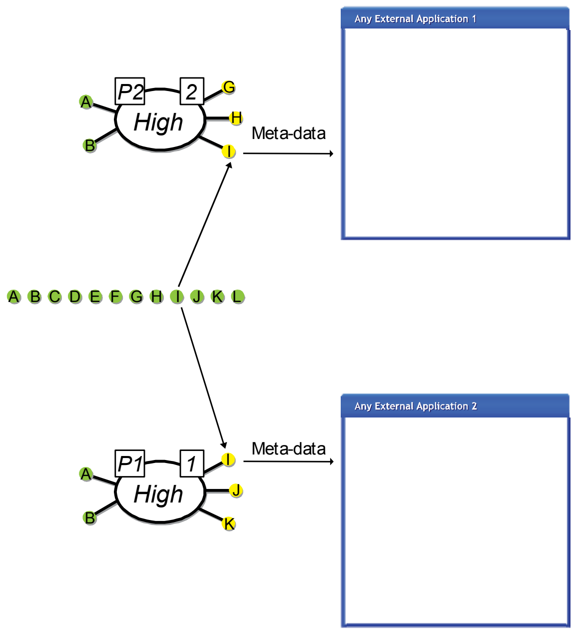


Figure 21: To make it possible to define multiple knowledge objects pointing to a single parameter, a common list of parameters is used to communicate values. However, each knowledge object carries instances of the parameters in the collection of input and output parameters. (Input parameters are to the left of the objects and outputs to the right.)

The approach of using a common parameter list could be compared to a blackboard-system with agents, see further Hopgood (2001) and Sriram (1997).

5.2.3 Knowledge adaptability

When a knowledge object writes a value to a parameter, the precision value of that knowledge object can be stored in the parameter in the common list. That will make a history of what precision a parameter value has. A rule is proposed saying that a knowledge object may only overwrite a parameter if its precision value is strictly bigger than the current precision value of that parameter in the common parameter list. The precision value of a knowledge object might change in different areas of the design space using formula. When using constraints together with precision values in the common list, the system will run in the following sequence:

Do until the conflict set is empty:

1. *List all triggered objects not violating any constraints, exclude solved objects. Sort the list by precision.*
2. *Execute the knowledge object with highest precision ("first come, first served" if several objects with the same precision exist).*
3. *Clear all output parameters in knowledge objects dependent on the outputs from the fired knowledge object. This can cause rules to fire more than once, but is done to make sure that the output with the highest precision is the final results.*

5.2.4 Solving rule conflicts

Two different situations may occur when allowing the knowledge base to contain multiple knowledge objects for a single phenomenon. Let us say that there are four triggered objects at one stage in the conflict set. Three of the objects contain knowledge about phenomenon P1 and one object deals with phenomenon P2 (see Figure 22). Object number one and two have been assigned a high precision value, object number three a medium precision, and number four has a low precision value. The design parameters A to D and I are known, but J to N are unknown. The selected object to run in this situation will be object number one because it has the highest precision value and was added to the system before object number two. (Since objects one and two have the same precision value, the "first come, first served" rule is applied.)

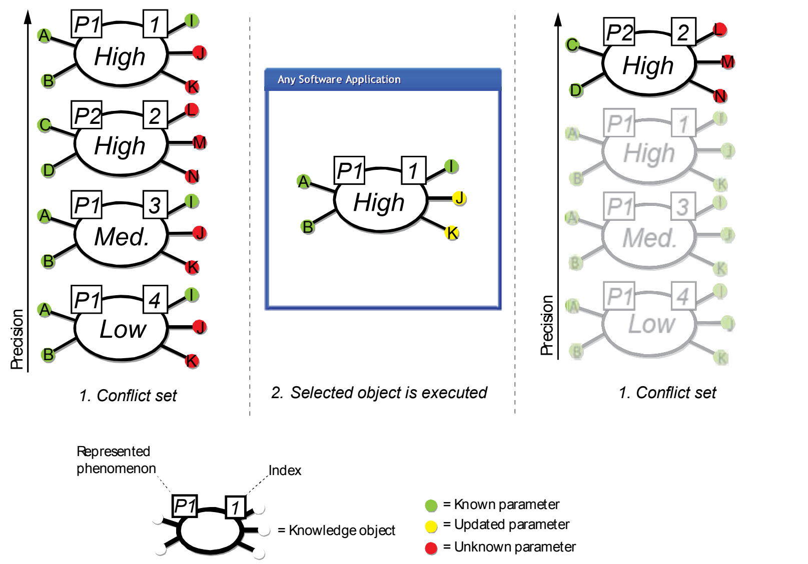


Figure 22: Objects do not overwrite values set by objects with precision values equal to or higher than their own precision value. (Input parameters are shown to the left of the objects and outputs to the right.)

Situation one: The parameter I was calculated by a knowledge object assigned a precision value equal to or higher than the precision value of object number one. Since knowledge object number one is selected, it will be executed using the pre-defined method. Values for parameter I to K will be calculated. But since an object with a precision value equal to or higher than the precision value of the current object (object number one) set the I parameter, the value of parameter I will not be overwritten. Knowledge object number one will set only the parameters J and K in this situation.

When updating the conflict set, knowledge objects 1, 3 and 4 will be considered solved since parameters I to K are known. The only object left in the conflict set is knowledge object number two.

Situation two: The parameter I was calculated by a knowledge object assigned a precision value smaller than the precision value of object number one. Since knowledge object number one is selected, it will be executed using the pre-defined method. Values for parameters I to K will be calculated. Since an object with a precision value smaller than the precision value of the current object (object number one) set the I parameter, the value of parameter I will be overwritten. All the parameters I to K will be set by knowledge object number one in this situation. Since the parameter I is changed, all dependent parameters must be cleared. When searching the knowledge base, it is found that knowledge objects 5 and 6 have the parameter I as input (see Figure 23). This will make the parameters O to T invalidated since they were calculated using the value of the I parameter with a smaller precision. When invalidating the parameters O to T, knowledge objects number 5 and 6 will be considered triggered and put to conflict set. Knowledge objects 1, 3 and 4 are considered solved.

At this stage, there are still four knowledge objects in the conflict set, so was the operation necessary? Yes, the value of the I parameter has a higher precision. Firing knowledge objects 5 and 6 using this better value will (probably) increase the precision of the values of the parameters O-T.

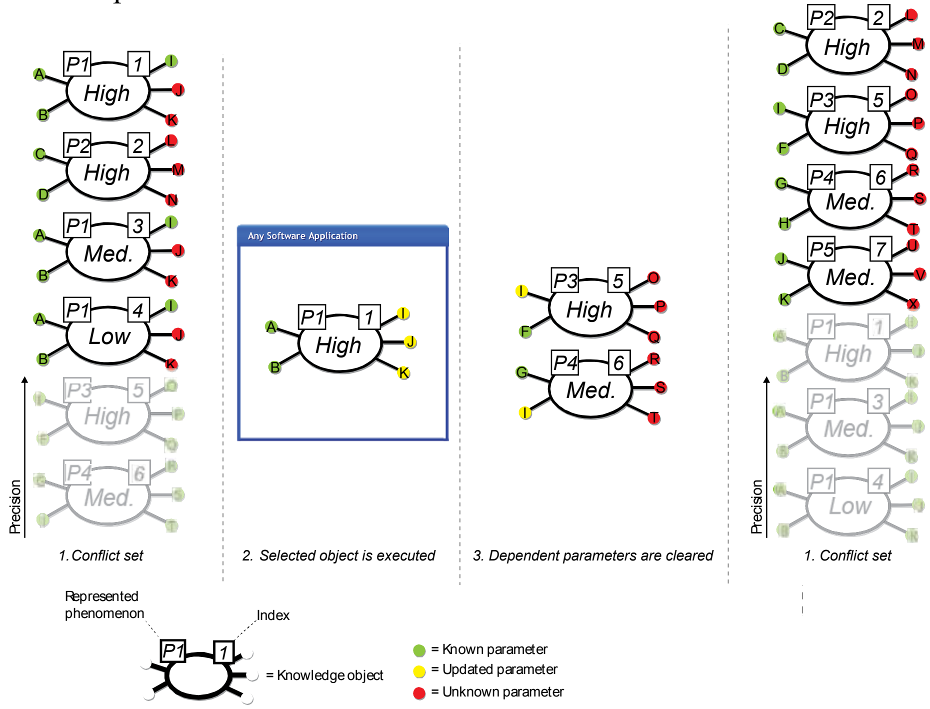


Figure 23: Objects do overwrite parameter values set by objects with precision value smaller than their own precision value. When that situation occurs, parameters dependent on the overwritten parameter are invalidated. (Input parameters are shown to the left of the objects and outputs to the right.)

5.2.5 Two types of meta-knowledge

Meta-knowledge can be implemented either in the inference engine or in the knowledge objects. The meta-knowledge implemented in the inference engine will be applied to every knowledge object. Take, for example, the rule saying that the knowledge object with the highest precision value should be selected to execute. That is a global rule and is suitable to be implemented in the inference engine. Other types of meta-knowledge are suitable to implement in the knowledge objects. It is, for example, often told that rules of thumbs are applicable only when the tube is “thin-walled”, often meaning that the wall-factor is greater than 10. Such meta-knowledge is best placed in the knowledge object itself.

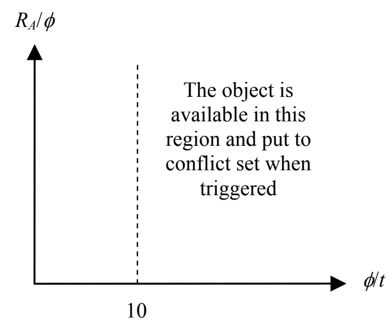


Figure 24: When implementing meta-knowledge in the knowledge objects, the system will know when different objects are applicable. In this example, the wall-factor needs to be greater than 10 to make the knowledge object available (known as the thin-wall assumption).

5.3 Evaluating the knowledge adaptability

In order to evaluate the knowledge adaptability, a knowledge base with conflicting knowledge objects was developed and solved, and the product preparation of the rotary draw bending was targeted. To speed the process of defining the knowledge objects and to make an overview of the solve process, a graphical user interface was developed. The user interface contains two main controls, one tree view showing the knowledge objects and parameters, and one frame showing the conflict set (see Figure 25). Functions for developing and maintaining the knowledge base were also developed and added to the graphical user interface.

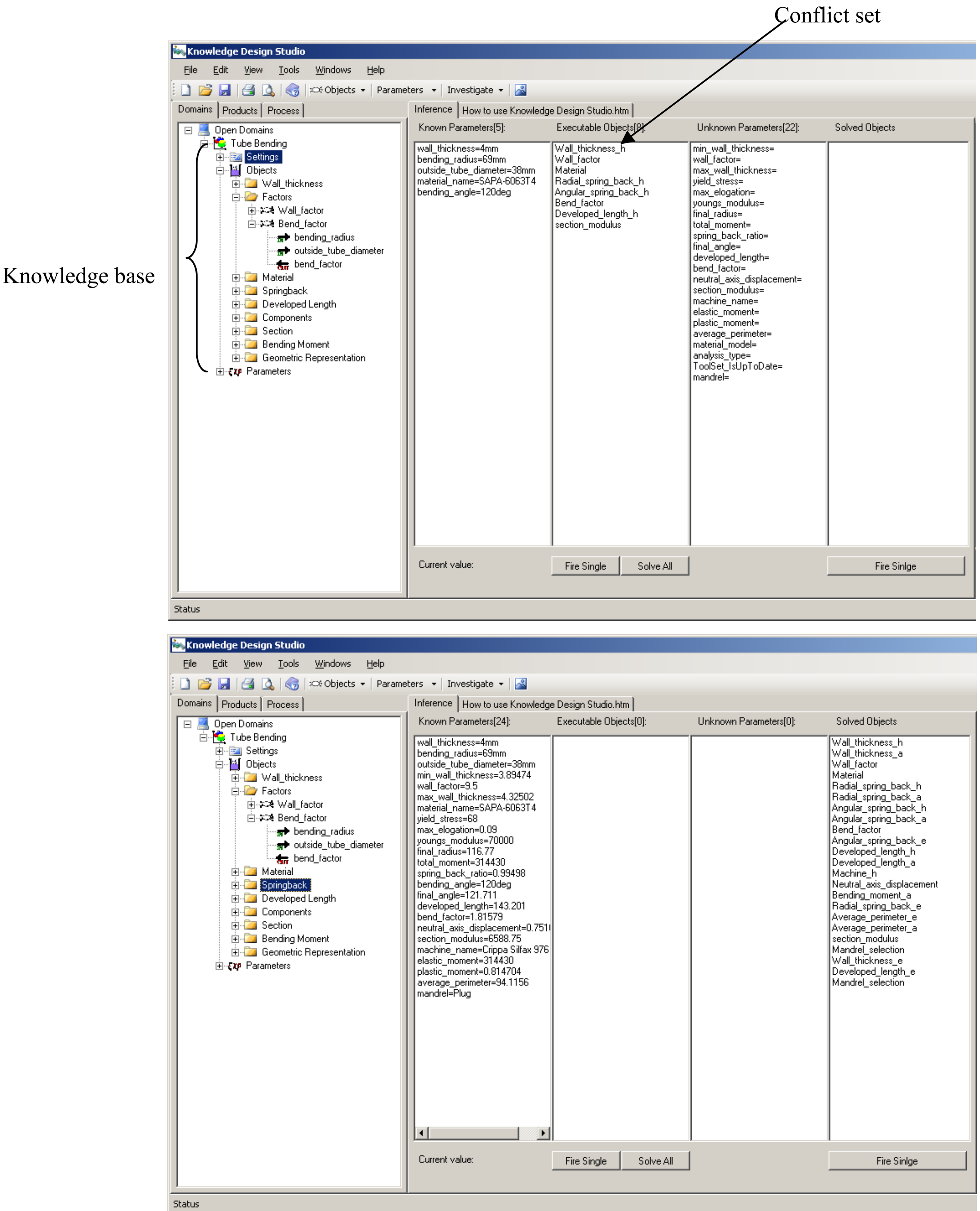


Figure 25: To make it easier to develop and maintain the knowledge base and to use the inference engine, a simple graphical user interface was developed. In the top picture, the knowledge base for the rotary draw bending is shown before running. In the bottom picture, the system has finished running.

5.3.1 Defining objects and parameters

The knowledge base implemented as a test is described in Appendix B. A parameter is viewed as an information carrier and in the common parameter list (see Figure 21) where there is no separation between input and output parameters. A sub-set of the common parameters implemented is listed in

Table 4, found in the Appendix A. Some additional parameters were added in order to make file transfer possible when implementing the FEA (for further information, see Appended Paper 3). In Table 5, found in Appendix A, the complete collection of implemented knowledge objects is listed, with input parameters, output parameters and the method is used by the knowledge objects.

5.3.2 Solving the knowledge base

In order to evaluate the system using the knowledge base with conflicting knowledge, it was run twice with the same input parameters but with different knowledge objects active in the knowledge base. In the first run, all the knowledge objects were active. In the second run, the knowledge objects using experimentally developed data were suppressed. In Table 2, the first run is listed. The user provided the parameter values found at iteration zero, and they form the product specification. The conflict set is updated with executable knowledge objects from which one is selected to execute (written with italic letters in the table). In the next iteration, new parameter values are available and the conflict set is again updated (objects already available are not listed twice in the tables if not selected for execution).

The execution sequence for the second run is listed in Table 3. That execution sequence is similar to when product specifications are given in a region of the design space where no experimentally developed knowledge is available. When comparing the two tables, it is clear that the use of experimentally developed knowledge is faster and that no FEM-calculations are generated to verify the design proposal.

Iteration	*Known parameters*	*Conflict set*
0	Wall_thickness=4mm Bending_radius=69mm Outside_tube_diameter=38mm Bending_angle=120deg Material_name=SAPA-6063T4	*Wall_factor* Material Bend_factor Section_modulus
1	Wall_factor=9.5	*Neutral_axis_displacement*
2	Neutral_axis_displacement=0.751611	*Material*
3	Yield_stress=68 Max_elongation=0.09 Young's_modulus=7000	*Radial_springback_e* Angular_springback_e Bending_moment_a Average_perimeter_e Wall_thickness_e Developed_length_e
4	Final_radius=67.84	*Developed_length_e*
5	Developed_length=140.1	*Average_perimeter_e*
6	Average_perimeter=91.6	*Wall_Thickness_e*
7	Min_wall_thickness=3.24 Max_wall_thickness=4.72	*Angular_springback_e*
8	Final_angle=116.77	*Bending_moment_a*
9	Elastic_Moment=314430 Plastic_Moment=0.81470 Total_Moment=314430	*Section_modulus*
10	Section_Modulus=6588.75	*Machine_h*
11	Machine=Silfax 76	*Mandrel_selection*
12	Mandrel=Plug	

Table 2: The run procedure when knowledge objects containing empirical data were enabled.

Iteration	*Known parameters*	*Conflict set*
0	Wall_thickness=4mm Bending_radius=69mm Outside_tube_diameter=38mm Bending_angle=120deg Material_name=SAPA-6063T4	*Wall_thickness_h* Wall_factor Material Radial_spring_back_h Angular_spring_back_h Bend_factor Developed_length_h Section_modulus
1	Min_wall_thickness=3.13636	*Wall_factor*
2	Wall_factor=9.5	*Wall_thickness_a* Material Neutral_axis_displacement Average_perimeter_a
3	Min_wall_thickness=3.89474 Max_wall_thickness=4.32502	*Neutral_axis_displacement*
4	Neutral_axis_displacement=0.751611	*Developed_length_a*
5	Developed_length=143.201	*Average_perimeter_a*
6	Average_perimeter=94.1156	*Material*
7	Yield_stress=68 Max_elongation=0.09 Young's_modulus=7000	*Bending_moment_a*
8	Total_moment=314430	*Radial_spring_back_a*
9	Final_radius=140.624	*Angular_spring_back_a*
10	Final_angle=121.711	*Bend_factor*
11	Bend_factor=1.81579	*Section_modulus* Mesh_size_h Material_model Mandrel_selection FE_Model_type
12	Section_modulus=6588.75	*Machine_h*
13	Machine_name=Crippa Silfax 976	*Mesh_size_h*
14	Mesh_size=3.31345	*Material_model*
15	Material_model=Piecewise	FE_Model_type
16	Analysis_type=Simple	*Mandrel_selection*
17	Mandrel=Plug	
	FEA_Prepared_CAD_File=c:\CAD\...	*Geometry_Update*
18	ToolSet_IsUpToDate=TRUE	
	Meshed_CAD_file= c:\CAD\...	*Mesh_Model*
19	FEA_IsUpToDate=TRUE	
	CATIA_Bulk_Mesh=c:\TEMP\...	*Mesh_Exports*
20	Bulk_IsUpToDate=TRUE	
	Mesh_Model=c:\FEM\...	*Mesh_Translation*
21	Mesh_IsUpToDate=TRUE	
	FEA_File=c:\FEM\...	*FEM_Control cards*
22	FEA_File_IsUpToDate=TRUE	
	Execution_Folder=c:\FEM\Run1\	*FEM_Calculation*
23	FEA_Executed=TRUE	*Wall_Thickness_n*
	Wall_thickness_Max=4.674 Wall_thickness_Min=3.514	

Table 3: The run procedure when no knowledge objects containing empirical data were enabled.

CHAPTER 6

FLEXIBLE AUTOMATION OF THE FINITE ELEMENT ANALYSIS PROCESS

The process of setting up, running, and post-processing a finite element analysis (FEA) can be a subject for automation within a design automation system as a local system. In this chapter, how to make use of the KBE-system to automate FEA is shown. The automation of FEA is applicable when similar calculations are performed often, as is true when designing toolsets for the rotary draw bending.

6.1 Running an automated FEA-process within the global KBE-system

Automating the FEA-process and adding the supporting methods to the KBE-system is the same as adding a local automation system into the global design automation system. The execution process of the global design automation system will then be as shown in Figure 26. What starts the process is the presentation of product specifications. This will trigger the inference engine to execute directly available knowledge objects, ending up with a design proposal. The design proposal is represented in a CAD-system where idealizations are defined as well. The design representation with its idealizations is interpreted to generate a FE-model. Knowledge objects in the global design system are used to generate a recipe for how to execute the FE-model. The FE-model and the recipe are sent to a FEM-solver. When the solver has finished, other knowledge objects in the global design system interpret the results by evaluating whether the design proposal was good. The different steps are described in detail in the subsections below.

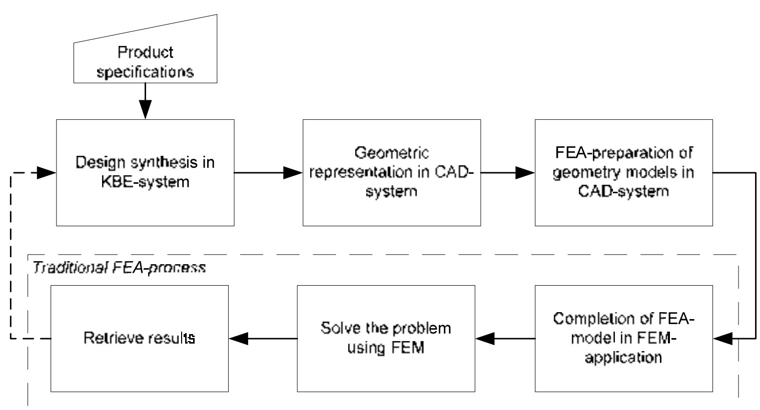


Figure 26: The execution process of a KBE-system using an automated system of the FEA-process as a local design automation system, modelled as knowledge objects.

6.1.1 Geometric representation

The geometric representation of the design proposal can be established by a generative or a template approach. The former approach is usually achieved by using macro programming in the CAD-system, while the latter uses pre-defined CAD-models that are configured. Combinations of the two approaches are possible.

One benefit of the template approach is user friendliness. It is easy to make pre-defined models using feature-based modelling in a modern CAD-application and make the system do the configuration. One drawback with the template approach is that it can be hard to do the design automation systems general.

6.1.2 FEA-preparation

When automating the FEA-process, the geometrical representations of the product (the CAD-models) need to be prepared for the automated discretisation of the problem domain. One approach to doing the preparation is to let skilled computation-engineers create publications* in the CAD-files that tell the system how to create different idealizations of the problem in different areas of the design space. Simple scripts or macros in the CAD-system can support this procedure.

When using publications in the CAD-models as a base for the automatic meshing method, the mesh can be generated either by the CAD-systems built-in meshing functions or by external software with such functions.

6.1.3 Completion of the FEA-model

The calculation engineer often defines the mesh size of a calculation model based on his/her experiences. In order to automate the FEA process, this step also has to be automated. Two approaches are suggested here. They are to either apply a heuristic rule or do experiments in the extremities of the design space. Input to the process is design parameters from specifications or calculations performed by the top-level KBE-system. Output is either a global value for the mesh size or local values for sub systems or regions of components. When it comes to material models, quite a big number exist to select from, and it is not always obvious which one is the best. In this case, the FEA-specialist needs to do serious investigations and evaluations in order to select and develop correct material models and rules saying when to use them. This is important continuous work the automation of FEA is intended to free the FEA-specialist to do. Input to the method for selecting material model is design parameters from specifications or calculations performed by the top-level KBE-system. The output of the completion of the FEA-model process is a recipe of how the FEM-calculation should be performed by the solver, commonly referred to as control cards.

* Using publications in the CAD-system is similar to encapsulation in the object-orientated programming paradigm, but applied to geometry. Publications make it easier to communicate geometry in the product structure telling what geometry it is allowed to make references to.

6.1.4 Running the simulation

Usually it is possible to execute a FEM calculation from the command prompt. When that is the fact, it is also possible to execute the calculation from a single line in a program using a "shell" function. Input for the execution object is the input file and environmental values. The output is the output databases file and a Boolean value telling whether the run was successful.

6.1.5 Retrieving results

The FEM-applications investigated by the author are LS-DYNA and ABAQUS. These two applications generate outputs that either are ASCII based or interpretable by phyton-scripts. Creating simple subroutines, it is possible to retrieve common results of interests such as maximum stress, displacement or plastic strain. Input for these objects is the output file, and output from the object is values for interesting parameters. An object retrieving all interesting values at once or several objects retrieving one or a few of the interesting values can be developed. The latter tends to be more flexible.

6.2 Evaluating the automation of FEA

To automate the FEA, routines were developed and added to the implemented system (see Section 5.3). The routines were added as methods available to the inference engine through the connection layer (see Figure 14). The toolsets of the rotary draw bending were then targeted for the FEA automation. The developed routines were used to do an extensive exploration of a part of the design space. Sixteen points were selected in the range shown in Figure 27. For each point, several calculations were completed changing mesh size, product structure and material model, as shown in the figure. In Figure 28, four examples are presented. A more detailed description of the exploration of this part of the design space is found in (Aniket C. and Girish V., 2008). The calculations ran without failure.

6.2.1 Comparing to measurements

Even though the scope of this thesis work is not to develop methods for doing sound FEM-calculations, but rather to develop methods for design automation systems, it is interesting to compare FEM-results to reality. In Graph 1, the measured and the calculated spring back using FEM are plotted. It is clear that the calculated spring backs are not equal to the measured ones. One reason for this is that the toolsets are flexible in reality (since nylon tools were used). That was not captured in the models. Another phenomenon is that the aluminium material in the tubes is not isotropic due to the extrusion process. That phenomenon was captured in one of the calculations (plotted as Anisotropic in the graph). A general rule is that the more complex the FEM-model gets, the more time it will take to run. For example, when adding the anisotropic material behaviour, the run time was increased 20 times compared to when not taking that phenomenon into consideration. Adding the flexible toolsets would also increase the computation time several times. Hence, it is necessary to investigate what phenomenon is the most dominant and which phenomena are not necessary to consider. In Appendix B the experimentally developed data is presented in detail.

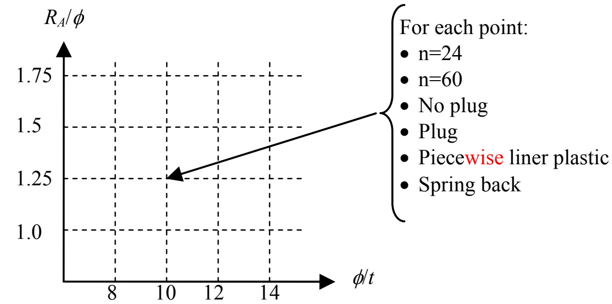


Figure 27. The investigated area of the design space.

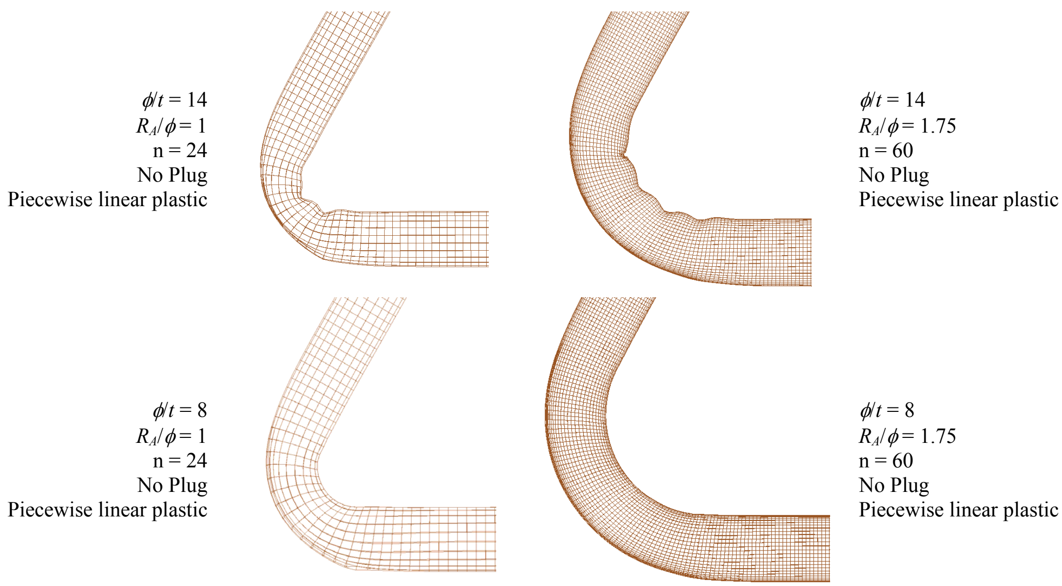
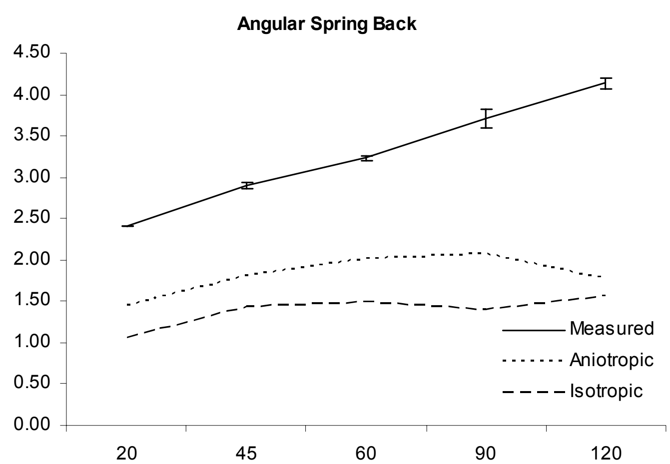


Figure 28. Example of output from the design space exploration done in order to test the system. The author supervised the running of the system in order to explore the design space. More details are found in (Aniket C. and Girish V., 2008).



Graph 1: The measured and calculated spring back plotted in the same diagram shows that it is not a easy task to capture all phenomena in a mathematical model (as FEM-calculations are).

CHAPTER 7

CONCLUSIONS

In this section, the research work presented in this thesis is concluded.

In this thesis, methods for automating the production preparation of rotary draw bending toolsets have been presented. In the introduction, an overall research question was stated and divided into three sub-questions. The answers are detailed below.

How to apply the concept of design automation to rotary draw bending toolsets for aluminum tubing? Experience dictates that, depending on the intended level of automation and intended flexibility of the automation system, it is possible to implement heuristic and analytical knowledge into KBE-systems using feed forward chaining and rule-based or object-based knowledge representation. A decision has to be made whether the system should allow multiple rules for a parameter (m to n mapping) or if a parameter may only be calculated from one rule (one to n mapping). If the m to n mapping is found beneficial, experiments have shown that object-oriented programming can be used to define knowledge objects as the knowledge representation and a search engine can be used to interpret the knowledge base in different situations.

It has proven to be possible to make the system able to analyze its design proposals through the automation of FEA-process for tube bending.

How should design automation systems be built to allow a high degree of flexibility? Using object-oriented programming has proven to be a fruitful way to make the design automation system highly flexible. Each rule or knowledge chunk is defined as an independent object with two lists containing inputs and outputs and an attached method for processing the inputs to outputs.

An object-oriented view of the product has proven to help make the design automation system flexible when creating geometric representations of the product.

When comparing CAD-integrated KBE-systems to stand-alone KBE-systems, the implementation will be more flexible with the stand-alone KBE-system if the knowledge base contains few geometrical relationships compared to the total number of rules and if the total number of rules is large compared to the number of geometric features in the geometrical representation. Otherwise, a CAD-integrated KBE-system allows more flexible implementations.

How should design automation systems be made able to handle the situation when multiple types of knowledge coexist for a single phenomenon? Experiments have proven that one solution is the implementation of constraints and precision values on the knowledge objects. This supports the selection of knowledge in different areas of the design space. Meta-rules can be put in the inference engine globally telling how to select knowledge in different situations and locally telling under what circumstances the knowledge is available and how accurate it is.

How may design automation systems be built enabling automatic but also flexible finite element analysis of the rotary draw bending of aluminium tubing? Experience says that prepared geometric representations can be used to support the automated mesh creation and that it is possible to use a global KBE-system to make a recipe for how to run the FEM-calculation. The FEM-calculations are then executed and interpreted automatically using the KBE-system.

CHAPTER 8

DISCUSSION AND FUTURE WORK

In this chapter, the research work presented in this thesis is discussed.

8.1 How general are the methods presented in this thesis work?

When conducting research in the field of design automation, one finds oneself being part of three processes: the research work, the building of the design system, and the study of the problem domain. Although it sometimes has been difficult, the design automation in general has been the main focus during the research work carried out in this thesis. When reviewing the answers to the research questions, they actually seem to be general enough to be applied on a range of production processes. Still, they have only been tested on one production process and, hence, future trials have to be done.

Applying the methods presented in the thesis when building design automation systems will make the resulting system flexible to use and maintain. It will also make the knowledge base able to contain conflicting knowledge. When using knowledge bases allowing confliction knowledge, it is possible to implement knowledge used in different phases of the product development process in a single system. Another benefit is the automation of FEA-process for the rotary draw bending (that can be adapted to other production processes) which makes the system able to analyze the generated design proposals.

8.2 Different research cycles

The first iteration of the empirical cycle included the development of a design automation system for rotary draw bending using a readily available software application developed to do that kind of task, presented in Appended Paper 1. During that development phase, some questions arose: What about the different kinds of knowledge? What about automatic FEA? These questions became the research questions in this research work.

Questions come from practice, and answers from theory. Further, when building theory, it is critical to be practical. In this work, the most interesting questions answered in this thesis arose after the first research cycle. Yet the second cycle raised even more interesting questions. When I reflect over my research work so far, I realize that I have acquired a lot of experience in three domains: how to conduct research, how to build design automation systems, and how to design toolsets for the rotary draw bending. The experience acquisition process has been described by Groot (1969) and is depicted in Figure 29.

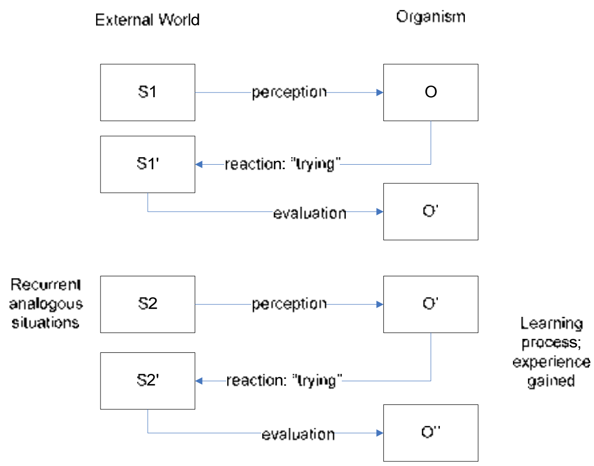


Figure 29: S1 and S2 are similar situations. O is the organism (me as researcher). (Groot, 1969) depicts the acquisition of experience.

Since there are three different domains I have gained new experiences in and each affects my research in design automation, it is clear that I need competence in all of them to do research in design automation. In Figure 30, two diagrams are shown where the different competencies needed are plotted on a relative scale. In both of them, the two cycles presented in this thesis are plotted (this is, of course, just subjective estimations). What happens if I would perform a new research cycle on design automation systems for the rotary draw bending? Or what if I did a new research cycle on design automation system for some other domain?

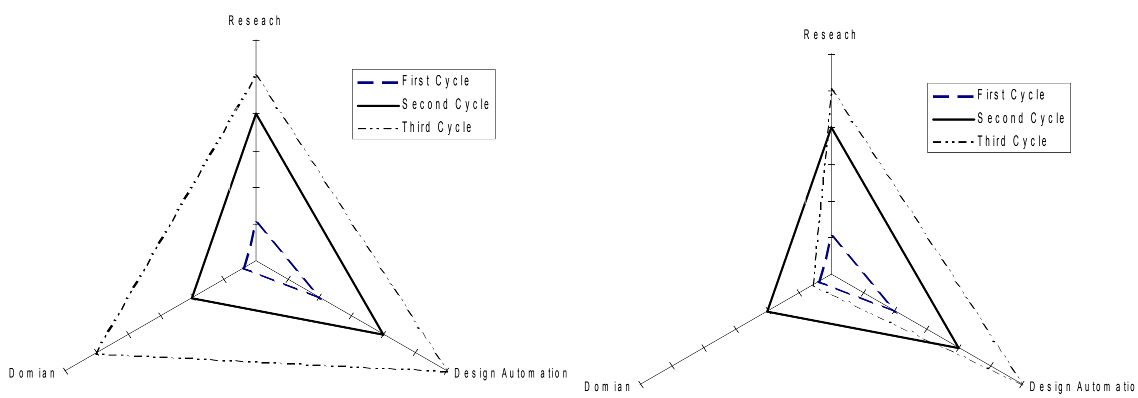


Figure 30: The quality of the research conducted in the field of design automation is dependent on three competencies: research competency, design automation competency, and specific domain competency.

The second iteration of the empirical cycle included the development of a design automation system for rotary draw bending using not yet developed software able to solve these kinds of tasks. In that case, computer programming can also be added as a competency in the diagrams.

8.3 Verification

Through experiment it has been verified that the concept of design automation can be applied on rotary draw bending toolsets using either CAD-integrated or stand-alone KBE-systems. Experienced engineers that have been designing toolsets for the rotary draw bending for many years have been part of the project and have had insight in to the knowledge base implemented including the heuristic, analytical, numerical and experimental knowledge. Secondly have the system been built twice, first using a readily available software using forward chaining and production rules. And then using knowledge objects to represent the knowledge base and using a forward chaining mechanism to search the knowledge base. Since the two implementations have proven to work within their scope the results are viewed as empirically verified.

8.3.1 The flexibility of the design automation systems

In this thesis, it is stated that the flexibility of a design automation system will increase if the object-oriented view is adapted to the knowledge base. This has been verified through the implementation of an inference engine and a knowledge base using the object orientation. In that implementation, it is possible to create, edit, suppress, and remove knowledge chunks without breaking the system. The operations can be done regardless of what order the knowledge was added in (i.e. there is no history). When editing the knowledge base, the execution sequence will change automatically according to changes. The drawback with the system being that flexible is that it is hard to foresee what effects changes in the knowledge base will have.

There are two options on where to put the knowledge base when implementing a design automation system. Either the knowledge base is built into the product structure using a CAD-integrated system or the knowledge base is built in a stand-alone application. Through experiments it is proven that a stand-alone KBE-system is suitable when the number of geometrical relationships is low compared to the total number of rules in the knowledge base and when the number of geometry features is low compared to the total number of rules. Otherwise, a CAD-integrated system is suitable for the implementation of the knowledge base.

If the design automation system is integrated with a CAD-system, the flexibility is said to be increased though the object-oriented view on the product structure as well. The object-oriented view of the product structure makes the systems flexible so that they may edit and suppress product components without breaking the system. It has been verified that the object-oriented approach to the product structure increases the flexibility of the system. It does so by implementing both a system where macros are used to achieve the object orientation and a system that uses publications and external links.

8.3.2 Multiple rules for one phenomenon

It has been experimentally verified that it is possible to implement multiple rules for a single phenomenon using knowledge objects, a common parameter list, and precision values. The implemented system has 38 knowledge objects concerning ten different phenomenon or design tasks. Different objects will be executed, depending on what inputs are presented to the system. One drawback is that it is hard to foresee the execution sequence of the system. It is also hard to get an overview of where in the design space knowledge objects overlap, and where they exist at all.

8.3.3 Automation of the analysis process

It has been verified through experiments that it is possible to automate the finite element analysis process for rotary draw bending. The FEA-model is automatically generated, executed, and interpreted by the system. The automation steps were implemented using knowledge objects. Since the computer routines used by these knowledge objects were specially developed for the experiment, they might not apply to other production processes than the rotary draw bending. The methods used to implement the routines are general enough to be used on other processes.

One drawback of including the analysis process into the KBE-system is the long execution time. There should be a warning to the user before starting the analysis process.

8.4 Validation

When running the implemented systems, it is realized that design automation has a large potential to cut lead-time. It takes only a couple of minutes to generate a design proposal and approximately an hour to verify the proposal using the automated FEA-process. This should be compared to the manual process that may take up to a day or more. Within the domain of circular sections, the design automation system makes the adaption of different costumer specifications fully automated.

One benefit of the implementation of experimentally developed knowledge is that the system will be faster to execute as the database of experiments is increased over time if production outcome is fed back into the system.

The research and systems implemented are restricted to circular profiles.

8.5 Future work

Finally, the research presented in this thesis raises many new research questions:

- *How can one loop the system when having several bends on a tube?*
- *How can one extend the application to general closed and open profiles?*
- *How can one make the system a closed synthesis/analysis loop allowing for optimisation against customer specified product tolerances?*
- *Is it possible to define multiple methods in a knowledge object to make the system able to execute dynamically, depending on presented input parameters?*
- *Is it possible to make use of the conflicting knowledge to have better design proposals?*

APPENDIX A

IMPLEMENTATION OF THE KNOWLEDGE BASE

This appendix contains detailed tables of used parameters and implemented knowledge objects. All parameters used in the knowledge base are listed in Table 4. Table 5 includes all knowledge objects implemented in the knowledge base.

Table 4: The parameter list used in the knowledge base. Additional parameters were added in order to handle different files for geometric representations.

Name	*Description*
Bending_angle	Defines the nominal bending angle
Bending_radius	Defines the nominal bending radius
Bending_factor	The bend factor
Average_perimeter	The final average perimeter
Developed_length	The developed length
Elastic_moment	The moment necessary to make the material plastic
Final_angle	The final angle after releasing the clamp
Final_radius	The final radius after releasing the clamp
Machine_name	The name of the selected bending machine
Mandrel	What type of mandrel should be used, empty, plug, flexible (with regular, close or ultra close pitch)
Material_name	The name of the tube material
Max_elongation	Max elongation of the material
Max_wall_thickness	Maximum wall thickness after bending
Min_wall_thickness	Minimum wall thickness after bending
Neutral_axis_displacement	The displacement of the neutral axis of the tube section due to changes in the wall thickness
Outside_tube_diamter	Defines the outside tube diameter
Plastic_moment	The moment necessary to make the bend in the plastic region
Section_modulus	The modulus of the tube section
Spring_back_ratio	The ratio between nominal and final bending angle
Total_moment	The total moment needed to do the bending action
Wall_factor	The ratio between the outside tube diameter and the wall thickness
Wall_thickness	Defines the wall thickness of the tube
Yield_stress	The material yield stress of the tube
Youngs_modulus	The Young’s modulus of the tube material

Table 5: To model the knowledge base presented in Appendix B, 38 knowledge objects were defined containing heuristic, analytical, numerical and empirical knowledge. Some knowledge objects just deal with geometry (mesh included). The method can be external, commercial or in-house developed software applications, or it can be a local string for easy mathematical expressions.

Name	*Inputs*	*Outputs*	*Method*	*Description*
Angular_spring_back_a	Bending_angle Developed_length Bending_radius Spring_back_ratio	Final_angle	MathCAD	Calculates the angular spring back based on an analytical rule using MathCAD
Angular_spring_back_e	Youngs_modulus Total_moment Bending_radius Outside_tube_diameter Yield_stress Wall_thickness Wall_factor	Final_angle Spring_back_ratio	Excel	This object executes Excel to retrieve an experimentally developed value of the angular spring back.
Angular_spring_back_h	Bending_angle	Final_angle	MathCAD	Calculates the angular spring back based on a heuristic rule using MathCAD
Average_perimeter_a	Wall_thickness Wall_factor Outside_tube_diameter	Average_perimeter	MathCAD	Calculates the average perimeter based on an analytical rule using MathCAD
Average_perimeter_e	Youngs_modulus Total_moment Bending_radius Outside_tube_diameter Yield_stress Wall_thickness Wall_factor	Final_radius Spring_back_ratio	Excel	This object executes Excel to retrieve an experimentally developed value of the average perimeter
Bend_factor	Outside_tube_diameter Bending_radius	Bend_factor	Local	Calculates the bending factor
Bending_moment_a	Wall_thickness Outside_tube_diameter Yield_stress Bending_radius Young's_modulus Bending_angle	Elastic_moment Plastic_moment Total_moment	MathCAD	Calculates the elastic, plastic and total bending moment necessary to do the bending. The calculation is based on an analytical knowledge implemented using MathCAD
Bending_moment_n	FEA_Output_file FEA_ran_without_failure	Total_moment	Developed by the author	Retrieves the maximum bending moment needed to do the bending. This value is found in the output files for the constant rotation boundary
Developed_length_a	Bending_angle Bending_radius Neutral_axis_displacement	Developed_length	MathCAD	Calculates the developed length based on an analytical rule using MathCAD
Developed_length_e	Youngs_modulus Total_moment Bending_radius Outside_tube_diameter Yield_stress Wall_thickness Wall_factor	Final_length	Excel	This object executes Excel to retrieve an experimentally developed value of the developed length
Developed_length_h	Bending_angle Bending_radius	Developed_length	MathCAD	Calculates the developed length based on a heuristic rule using MathCAD
FE_model_type	Wall_factor Bend_factor	Analysis_type	Excel	Selects appropriate level of idealizations to use when doing the discretisation of the problem when preparing the FEM-model.

Name	*Inputs*	*Outputs*	*Method*	*Description*
FEA_completion	Material_model FEA-Model Mesh_model	FEA_Model_IsUpToDate	Developed by the author	Adds control cards to the mesh file to make the FEM-model ready to run
Geometric_representation	Wall_thickness Outside_tube_diameter Bending_radius Machine_name Mandrel FEA_prepared_CAD_file	ToolSet_IsUpToDate	Developed by the author	Updates the geometric representation to current parameter values
Machine_h	Section_modulus	Machine_name	Excel	Selects what machine to use to bend the tube based on heuristic knowledge
Mandrel_selection	Bend_factor Wall_factor	Mandrel	Excel	Selects what type of mandrel to use when bending the tube based on heuristic knowledge
Material	Material_name	Yield_stress Youngs_modulus Max_elongation	Excel	Retrieves material data using the material name.
Material_model	Wall_factor Bend_factor	Material_model	Excel	Selects appropriate material model to use in the FEM-calculations based on heuristic knowledge
Mesh_export	Meshed_CAD_file FEA_IsUpToDate CATIA_Bulk_Mesh	Bulk_IsUpToDate	Developed by the author	Exports the CATIA-mesh model to a file
Mesh_size_h	Wall_factor Bend_factor	Mesh_size	MathCAD	Calculates the mesh size based on a *proposed* heuristic rule using MathCAD
Mesh_the_model	FEA_prepared_CAD_file Mesh_size Meshed_CAD_file Analysis_type ToolSet_IsUpToDate	FEA_IsUpToDate	Developed by the author	Creates a mesh model based on the geometric representation
Mesh_translation	Bulk_IsUpToDate Mesh_model CATIA_Bulk_Mesh	Mesh_IsUpToDate	Developed by the author	Translates the CATIA mesh to LS-Dyna mesh.
Neutral_axis_displacement	Wall_thickness Wall_factor Outside_tube_diameter Bending_radius	Neutral_axis_displacement	MathCAD	Calculates the neutral axis displacement based on an analytical rule using MathCAD
Radial_spring_back_a	Youngs_modulus Total_moment Bending_radius Outside_tube_diameter Yield_stress Wall_thickness Wall_factor	Final_radius Spring_back_ratio	MathCAD	Executes MathCAD to calculate the final radius based on an analytical rule
Radial_spring_back_e	Bending_radius Outside_tube_diameter Bending_angle Youngs_modulus Wall_thickness	Final_radius	Excel	This object executes Excel to retrieve an experimentally developed value of the radial spring back.
Radial_spring_back_h	Bending_radius	Final_radius	MathCAD	Executes MathCAD to calculate the final radius based on a heuristic rule
Section_modulus	Outside_tube_diameter Wall_thickness	Section_modulus	MathCAD	Calculates the section modulus using MathCAD
Wall_factor	Outside_tube_diameter Wall_thickness	Wall_factor	Local	Calculates the wall factor
Wall_thickness_a	Wall_thickness Wall_factor	Min_wall_thickness Max_wall_thickness	MathCAD	This object executes MathCAD to retrieve an analytical value of the minimum and maximum wall thickness

Name	*Inputs*	*Outputs*	*Method*	*Description*
Wall_thickness_e	Bending_radius Outside_tube_diameter Bending_angle Youngs_modulus Wall_thickness	Min_wall_thickness Max_wall_thickness	Excel	This object executes Excel to retrieve an experimentally developed value of the minimum and maximum wall thickness
Wall_thickness_h	Wall_thickness Bending_radius Outside_tube_diameter	Min_wall_thickness	MathCAD	This object executes MathCAD to retrieve a heuristic value of the minimum wall thickness

APPENDIX B

THE ROTARY DRAW BENDING KNOWLEDGE BASE

It was mentioned in Chapter 3.2, radial and angular spring back, final wall thickness, perimeter change, and developed length need to be considered when preparing the production equipment for rotary draw bending. In this chapter, the spring back and wall thickness change are described in detail, and how different knowledge types coexist is shown.

B.1 Heuristic rules

When asking skilled engineers that have done production preparation for rotary draw bending for years about spring back and wall thickness change, the answer was similar to what is found in handbooks.

When the bend radius is small, the tool will be manufactured to the nominal bend radius. It will then be decreased during a trial-and-error process until a satisfactory bend radius is found.

An expression for the angular spring back is found by doing linear interpolation between two experiment bends, typically 20 degrees and 120 degrees. A simple software application has been developed to help that process (Cone, 2007).

The minimum wall thickness is usually calculated using the following formula (Bradley, 2007; Cone, 2007; Gillanders, 1994; Miller, 2003):

$$t_{\min} = \frac{R}{R + \frac{\phi}{2}} t \qquad (1)$$

B.2 Analytically derived rules

When searching the scientific literature, some interesting works on tube bending are found. These publications contain rules analytically derived from fundamental physical laws and theories on the plastic behaviour of materials. The rules found make the assumption of ideal plastic material behaviour.

To calculate the final radius of the tube after the bend action, the following rule is found (Abdel-Malek and Maropis, 1998; Gillanders, 1994; Miller, 2003):

$$\frac{R_{\mathrm{A}}}{R_{\Omega}} = 1 - \frac{M}{E \cdot r^3 \cdot t \cdot \pi} \qquad (2)$$

This expression assumes constant plastic regions over the whole bend. In other words, the transition zones are not considered. For bending angles above 90 degrees, this assumption is fair enough. However, when decreasing the bending angle, the spring back will be increased to such an extent that this approximation will fail. The reason is that the transition zones, where the wall thickness is not completely plastic, will play a large role (see figure 31).

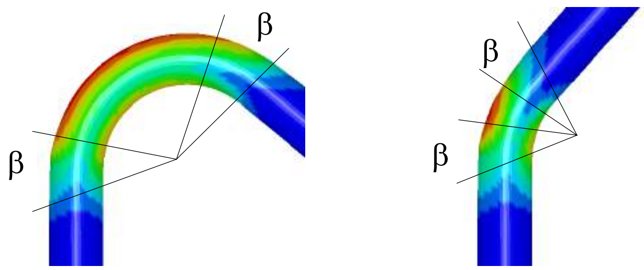


Figure 31: The transition zones account for a big part of the spring back at small bending angles.

To calculate the angular spring back, it is realized that the following geometrical relationship exists:

$$\frac{\theta_{\mathrm{A}}}{\theta_{\Omega}} = \frac{L_{\mathrm{A}}}{L_{\Omega}} \cdot \frac{R_{\Omega}}{R_{\mathrm{A}}} \tag{3}$$

To evaluate that expression, an estimation of the developed length has to be done. An expression for the developed length found in (Tang, 2000) is:

$$L = R_{\mathrm{A}} \cdot \theta_{\mathrm{A}} - 0.636r \frac{N_i - N_o}{N_i + N_o}(\theta_{\mathrm{A}} + \beta) \quad \text{, where} \tag{4}$$

$$N_o = \int_0^{\frac{\pi}{2}} \left(\frac{2k+1}{2k+2-\cos\alpha} \right) \times \left(1 - \frac{2k+\cos\alpha}{4k+3-\cos\alpha} \frac{\cos\alpha}{2k} \right) d\alpha$$

$$N_i = \int_{\frac{\pi}{2}}^{\pi} \left(\frac{2k+1}{2k+\cos\alpha} \right) \times \left(1 - \frac{2k+2-\cos\alpha}{4k+1+\cos\alpha} \frac{\cos\alpha}{2k} \right) d\alpha$$

The expression assumes elastic-perfectly-plastic behaviour, but also compensates for the transition zones (β) (see Figure 31). According to Tang, the value of β is about 30 degrees. However, that assumption is a rule of thumb, and no analytically derived expressions have been found.

To calculate the minimum and maximum wall thickness, the following formulas may be used (Tang, 2000):

$$t_{\min} = \left(1 - \frac{2k+1}{2k(4k+2)}\right)t \tag{5}$$

$$t_{\max} = \left(1 + \frac{2k+3}{3k^2}\right)t \tag{6}$$

This expression is more accurate than the heuristic one, and is not much harder to use than the expression presented as heuristic. In Tang's work, an expression for any point on the tube section is also given.

B.3 FEM calculations

In Chapter 6, the automation of the finite element analysis was described in detail. To support the design process, the FEM-models used to calculate the final wall thickness and the spring back were implemented as described in Chapter 6.2.

B.4 Experimental data

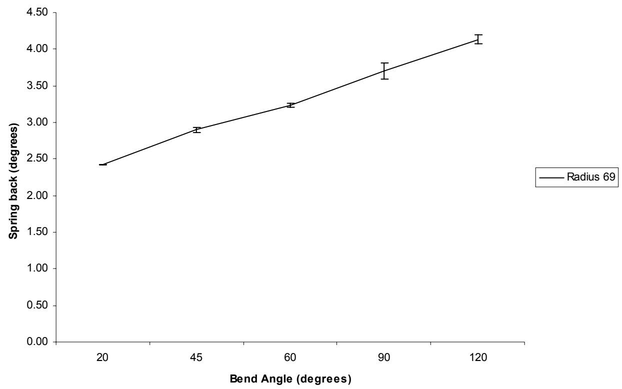
During the project, trial manufacturing was done and put into the knowledge base. Bending tubes made from three different materials (aluminium, steel and chromium-steel) developed the empirical data presented in this section. One bending radius was selected for the aluminium tubes, and three different bending radii were selected for the steel and chromium-steel tubes. The tubes were bent to five different angles. Five tubes were produced for each set-up. The different set-ups are listed in Table 6.

Material	*Aluminium 6063-T4*	*Steel 2394*	*Chromium-steel 17%*
Outer diameter (mm)	38	57	38
Wall thickness (mm)	4	1.5	1.5
Bend radius (mm)	69	60, 80, 100	39, 55, 65
Wall factor	1.82	1.05, 1.40, 1.75	1.03, 1.44, 1.71
Bend factor	9.5	38	25.3
Bend angle (degree)	20, 45, 60, 90, 120	20, 45, 60, 90, 120	20, 45, 60, 90, 120

Table 6: Trial manufacturing was conducted to make an experimentally based knowledge base. Five tubes were bent for each set up, making 175 tubes in total.

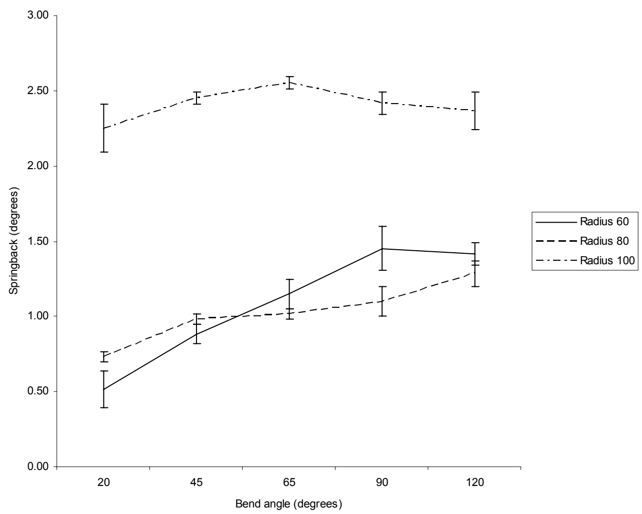
The final angles were measured using a measuring machine. The measured values of spring back are plotted in graphs 1-3.

Aluminium



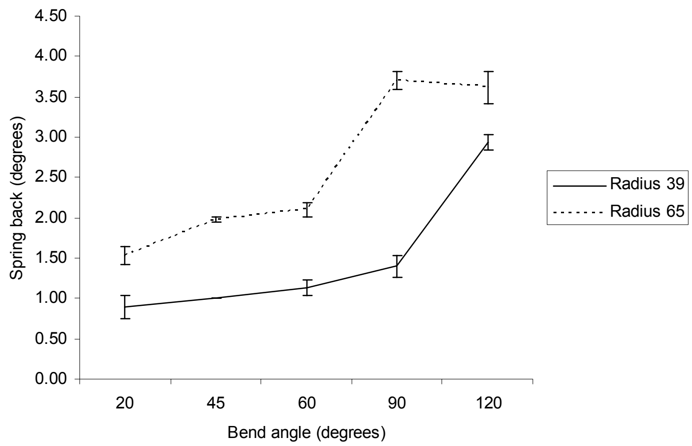
Graph 2: The spring back measured for one bend radius and five different bend angles.

Steel 2394



Graph 3: The spring back measured for three different bend radii and five different bend angles.

Chromium-Steel



Graph 4: The spring back measured for two different radii and five different bend angles.

REFERENCES

Abdel-Malek, K., and Maropis, N. (1998). A design-to-manufacture case study: Automatic design of post-fabrication mechanisms for tubular components. *Journal of Manufacturing Systems* **17**, 183-195.

Aniket C., P., and Girish V., C. (2008). Exploring the Design Space of Aluminum Tubing using Knowledge Objects and FEM. Master Thesis, Jönköping University, Jönköping.

Boart, P. (2007). The Enabling of Product Information in the Conceptual Phase, Luleå University of Technology, Luleå.

Bradley, D. (2007). Bend Tooling, Inc, Home Page. Vol. 2007. Bend Tooling, Inc.

C. Hernández, and Arjona, M. A. (2007). Design of distribution transformers based on a knowledge-based system and 2D finite elements. *Finite Elem. Anal. Des.* **43**, 659-665.

Catic, A., and Malmqvist, J. (2007). Towards Integration of KBE and PLM. *In* "sourceProceedings of ICED´07, Paris, France".

Cederfeldt, M. (2007). Planning Design Automation : A Structured Method and Supporting Tools, Chalmers tekniska högskola.

Chapman, C. B., and Pinfold, M. (2001). The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. *Adv. Eng. Softw.* **32**, 903-912.

Cone, M. (2007). Advanced Tubular Technologies, Home Page.

Elgh, F. (2007). Computer-Supported Design for Producibility : Principles and Models for System Realisation and Utilisation, Chalmers tekniska högskola.

French, M. J. (1999). "Conceptual design for engineers," 3./Ed. Springer, London.

Fulcher, J. (2006). Advances in applied artificial intelligence [electronic resource]. pp. xi, 309 p. Idea Group Pub., Hershey, PA.

Gillanders, J. (1994). "Pipe and Tube Bending Manual," 2/Ed. Rockford, Illinois.

Groot, A. D. D. (1969). "Methodology," Mouton & Co, Paris.

Hopgood, A. A. (2001). "Intelligent systems for engineers and scientists," 2./Ed. CRC Press, Boca Raton.

Jin, Z., Luo, S., and Daniel Fang, X. (2001). KBS-aided design of tube bending processes. *Engineering Applications of Artificial Intelligence* **14**, 599-606.

Jones, M. T. (2003). AI application programming [electronic resource]. pp. xix, 363 p. Charles River Media, Hingham, Mass.

Kaplan, A. M., and Haenlein, M. (2006). Toward a Parsimonious Definition of Traditional and Electronic Mass Customization. *Journal of Product Innovation Management* **23**, 168-182.

Luger, G. F. (2005). "Artificial intelligence : structures and strategies for complex problem solving," 5./Ed. Addison-Wesley, Harlow.

Löfstrand, M. (2007). A modelling and simulation approach for linking design activities to business decisions, Luleå tekniska universitet, Luleå.

Miller, G. (2003). "Tube Forming Processes: A Comprehensive Guide," Society of Manufacturing Engineers, Dearborn.

Roozenburg, N. F. M., and Eekels, J. (1995). "Product design : fundamentals and methods," Wiley, Chichester.

Sandberg, M. (2007). Design for Manufacturing - Methods and Pallications using Knowledge Engineering, Luleå University of Technology, Luleå.

Sellgren, U. (1999). Simulation-driven design, Royal Institute of Technology, Stockholm.

Simon, H. A. (1996). "The sciences of the artificial," 3./Ed. MIT Press, Cambridge, Mass.

Smith, P. (1988). Winning the New Products Rat Race. *Machine Design*, 95-98.

Smith, P. (1990). Fast-Cycle Product Development. *Engineering Management Journal* **2**, 11-16.

Sriram, R. D. (1997). "Intelligent systems for engineering : a knowledge-based approach," Springer, London.

Stokes, M., and MOKA Consortium (2001). "Managing engineering knowledge : MOKA: methodology for knowledge based engineering applications," Professional Engineering Publ., London.

Strano, M. (2005). Automatic tooling design for rotary draw bending of tubes. *The International Journal of Advanced Manufacturing Technology* **26**, 733-740.

Tang, N. C. (2000). Plastic-deformation analysis in tube bending. *International Journal of Pressure Vessels and Piping* **77**, 751-759.

APPENDED PAPERS

Three papers are appended to this thesis. They are appended in original format.

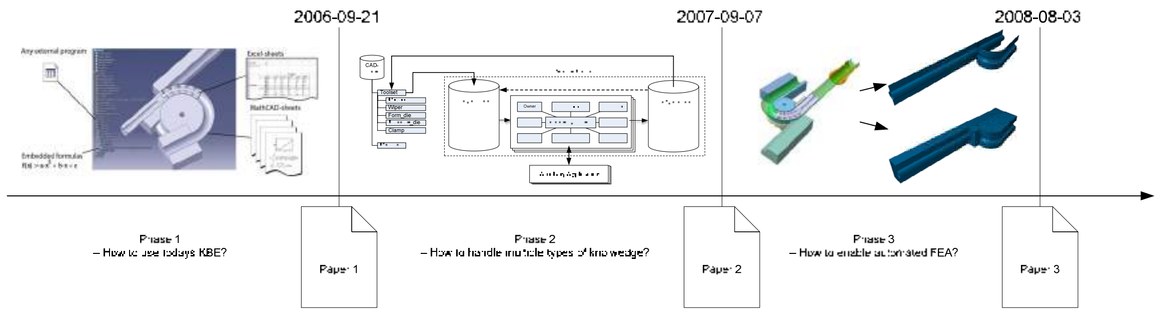


Figure 32: The evolution of the work over time. The last paper has been accepted and will be presented at ASME IDETC 2008 in August.

Paper A - ***Automated Design of Rotary Draw Bending Tools – An Approach Based on Generic CAD-Models Driven by Heuristic and Algorithmic Knowledge***

The first paper shows how a design automation system for the rotary draw bending of tubes can be built using a CAD-integrated KBE-system and connecting them to external software applications. It also describes the object-oriented approach when defining the geometric representation in CAD-systems.

Paper A is Copyrighted 2006 by EDMMP: Efficiency Development of Manufacturing Machines and Processes.

Paper B - ***A Flexible Design Automation System for Toolsets for the Rotary Draw Bending of Aluminium Tubes***

The second appended paper presents a method to allow multiple types of knowledge to co-exist for a single phenomenon. The method can be used to further enhance the flexibility of design automation systems. An implementation is presented in that paper as well showing explicitly how to model knowledge objects and how to use an inference engine to search the knowledge base.

Paper B is Copyrighted 2007 by ASME: Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Paper C - ***Manufacturability Analysis Using Integrated KBE, CAD and FEM*****. IDETC 2008**

The third appended paper presents a method for adding knowledge about how to execute and interpret FEM calculations. Implementing this method will yield highly potential design automation systems that cover a wide range of the design space.

Paper C is Copyrighted 2007 by ASME: Design Engineering Technical Conferences and Computers and Information in Engineering Conference.