MATEC Web of Conferences **21**, 05008 (2015)
DOI: 10.1051/matecconf/20152105008

Knowledge based system to support the design of tools for the HFQ forming process for aluminium-based products

Nikolaos Mekras[a] and Ioannis Artemakis

ANTER Ltd., Technology Development and Research Company, 11527 Athens, Greece

Abstract. In this paper a methodology and its respective implementation software system are presented that show how Knowledge Based Systems (KBS), both object-oriented and rule-based, when integrated with 3D CAD systems, can support the automatic design of tools for forming processes, and in our case for the solution heat treatment cold die forming and quenching process, known as HFQ TM[b]. Often, when designing tools (e.g. dies) for manufacturing processes, human experts' knowledge is used for decision making about final adjustments and improvements of the tools' designs and specifications. Using KBS methods and software tools, such process-specific and experts' knowledge can be organised, stored and used in a systematic way for obtaining results about tools designs adjustments and refinements. Besides the presentation of the proposed tools design methodology, a prototype software system is also presented that shows how the proposed methodology has been implemented for the automatic design and refinement of tools for the HFQ forming process for aluminium-based products.

1. Introduction

The design of manufacturing tools for forming processes is a complex task that uses as main input the geometry of the part that has to be manufactured and provides as output the geometry of the forming tools, including the geometries of the die, of the punch and of the blank holder. Additionally, the tools designer needs to consider information concerning the type of the material of the part, its thickness, the complexity of the part's shape (e.g. existence of vertical faces, sharp edges, etc.) and any other knowledge and information that originates from lab experience or from the shop floor. The way, in which this knowledge (part geometry, experts' rules, etc.) is gathered, organised, combined and used during the tools design work, influences the tools design results and the successful implementation of the manufacturing process. It is obvious that the tools design problem is in fact a quality problem for both the product and the tools, and it can be considered a complex problem, with several design parameters that may influence the final product quality. Especially the degree of the complexity of the shape of the product, adds complexity also at the forming tools design task which is based on: geometrical data processing, metals forming theory [3, 11], and also on requirements that arise from the

[a] Corresponding author: nmekras@anter-net1.com

[b] TM: HFQ is a registered trademark owned by Impression Technologies Ltd.

specific manufacturing process implementation. In this paper a methodology for the automatic design of forming tools is proposed, which is being developed for the HFQ (Hot Forming & Quench) process for aluminium based products within the EU FP7 Research project LOCOLITE (Project full title: "An industry system enabling the use of a patented materials processing technology for Low Cost forming of Lightweight structures for transportation industries").

The proposed methodology is implemented with the development of the software system KBES (Knowledge Based Engineering Design System), which integrates with the SOLIDWORKS CAD system and uses also Knowledge Based methods for object-oriented knowledge and rule-based experts' knowledge representation and processing. The KBES software is being also developed within the framework of the EU FP7 Research project LOCOLITE.

2. KBES functionalities

For many years Artificial Intelligence (AI) researchers have focused on efforts to represent knowledge and process it by computers trying to develop methods that resemble the way that humans, besides algorithms and mathematical procedures, store and process knowledge, especially expert knowledge e.g. in the form of rules or semantic relations. From these efforts several methods have appeared for the representation and processing of both rule-based and object-oriented Knowledge Representation Structures (KRS), that can also cooperate and work supplementary with algorithms and mathematical models. These knowledge representation and processing methods constitute the AI field of Knowledge Based Systems (KBS) [4]. For the development of the proposed KBES a combination of object-oriented and rule-based knowledge structures have been used, for representing both the basic geometry of the tools designs and the experts' knowledge for adjusting and optimizing these designs for the forming process under consideration.

2.1 Object-Oriented knowledge structures for tools geometry

The main concept of Object-Oriented (O-O) Knowledge Representation Structures (KRS) is that knowledge is organized in classes of entities, which are related and classified in the knowledge base. Each class is the general representation of the class objects, which use the class as their general template, and contains general information concerning class methods and class attributes [1, 4]. Each object of the class embodies and uses the class methods and attributes, where both are called class members and constitute the main elements in O-O languages and O-O programming in general. O-O programming and respective programming tools (e.g. C++, Java, C#, etc.) are used widely in nowadays in software engineering because of the better code organisation & objects classification, and of the improved code update and maintenance [2].

For the development of the KBES software, the O-O approach was chosen for the following reasons;

- The O-O knowledge representation is suitable for the representation and processing of geometric CAD objects concerning products and manufacturing tools [7, 10] and the update and maintenance of the respective code, and this is the reason why several 3D CAD systems (e.g. SOLIDWORKS) have been developed with the use of O-O methodology and O-O programming tools (e.g. C++, O-O .NET languages, etc.).
- A 3D CAD system had to be chosen as core CAD system, with which the KBES will be integrated for importing the product geometrical data, which are used as inputs for the tools geometry creation. For our case, SOLIDWORKS [12] was chosen because of its O-O features and possibilities for external code integration through its API (Application Programming Interface) functionalities.

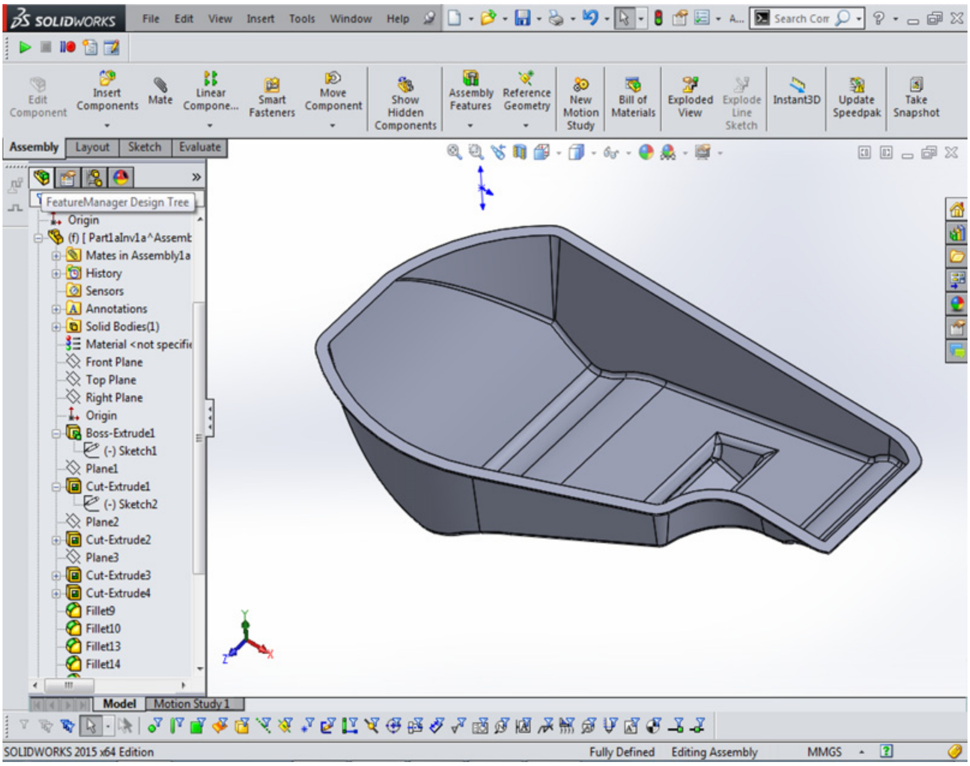


Figure 1. Object Oriented geometrical entities representation in the SOLIDWORKS CAD system.

An example of the O-O approach that SOLIDWORKS uses for CAD geometrical entities representation is given in Fig. 1. As it can be seen in the left frame of the SOLIDWORKS screen in Fig. 1, the Feature Manager Design Tree is showing in a tree structure the geometrical entities (features) of a part, which are organised using the O-O approach. The processing of these entities (e.g. geometry update, measuring, etc.) is also performed through the respective O-O methods of SOLIDWORKS, which make these compatible with the O-O external code of the KBES through the SOLIDWORKS API. The 3D design of the part, which is shown in Fig. 1, was created in SOLIDWORKS, and was used as one of our first case examples for testing the KBES functionalities for the automatic creation of the forming tools for the specific part.

2.2 Rule-based knowledge structures for process-specific & experts' rules

The Rule-based approach for knowledge representation was one of the first approaches of AI for gathering, organising and processing experts' knowledge for domain specific applications [4, 5]. Rules and their update mechanisms were the main elements of Expert Systems applications that appeared since the early 80s, with successful applications in several domains (e.g. engineering, medicine, etc.). In parallel, rule-based programming tools (e.g. the so called Expert System Shells) and programming languages were developed in an effort to support Expert Systems development [8] and to formalize rule-based knowledge representation and processing. A typical example of such tools is the programming language Prolog (Programming in Logic) [6, 9], which was initially developed as a language to solve problems of mathematical logic and which is very effective on rules representation and processing.

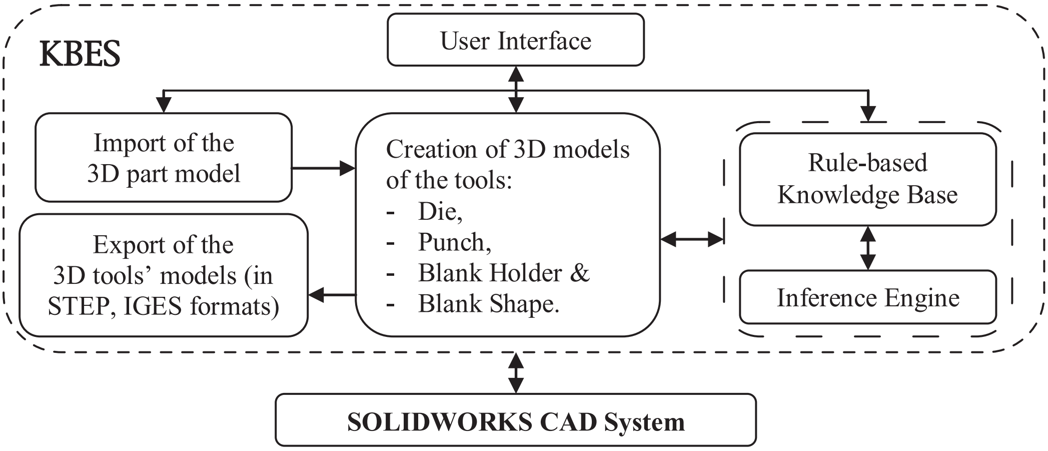

KBES
User Interface
Import of the 3D part model
Export of the 3D tools' models (in STEP, IGES formats)
Creation of 3D models of the tools:
- Die,
- Punch,
- Blank Holder &
- Blank Shape.
Rule-based Knowledge Base
Inference Engine
SOLIDWORKS CAD System

Figure 2. KBES functionalities diagram.

For adding and processing experts' rules within the KBES, the Prolog language was also used and was integrated with the rest KBES code, in an effort to organise and process better the rule-based knowledge for tools design.

2.3 KBES architecture & functionalities

The KBES basic architecture and main functionalities are presented in summary in Fig. 2:

As shown in Fig. 2, the KBES includes a set of functionalities for the automatic design of the tools after the 3D model of the product part is imported in the system. In the up to now version of the software, the imported file is in the SOLIDWORKS 2015 part file format (.sldprt), which can be read and recognised by the KBES, so that the automatic design of the forming tools can start.

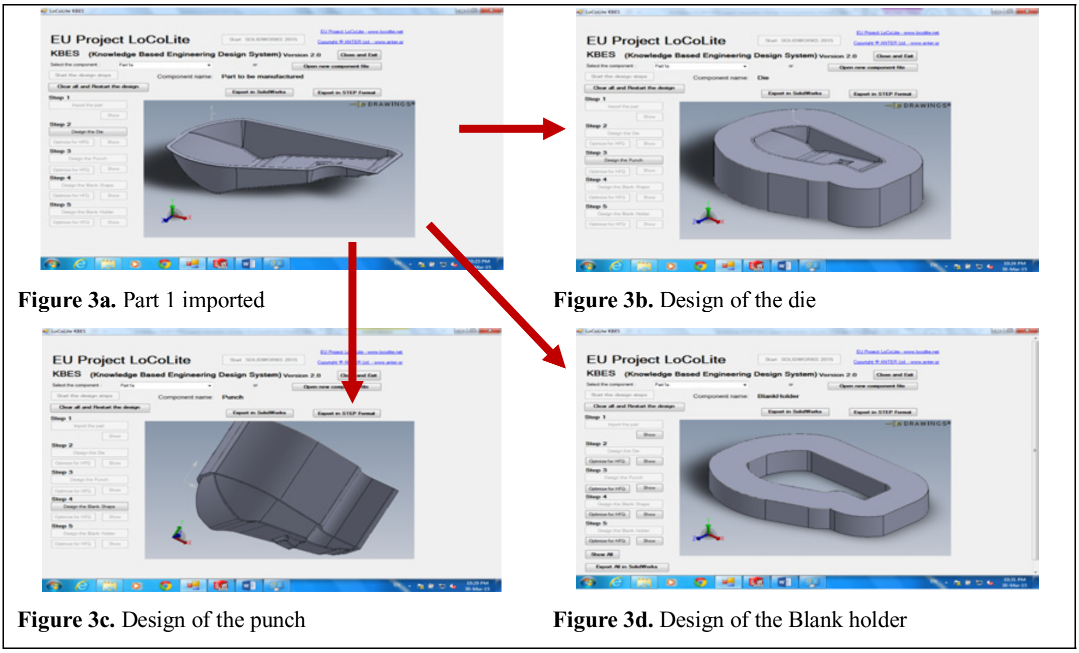
After the basic 3D designs of the main tools are automatically created by the system, including the die, the punch and the blank holder, the user can intervene and start making adjustments to the 3D tools shapes with the help of the rule-based knowledge base. This is a semi-automatic process, where the user can indicate which modifications are needed for adjusting and improving the 3D designs of the forming tools and use also the respective experts' rules for these modifications. For example the user can decide to: increase the contact area of the blank holder with the blank sheet and the die, increase the overall size of the die, increase the curvature of an edge of the die, or can adjust the slope of a non-vertical face of the punch, etc. After the user, decides and proposes to the system several adjustments for the tools' geometry, the KBES, with the help of the rules, re-designs again automatically the forming tools, so that the conditions of the rules are satisfied. For example a die design rule is based on the following condition:

$$\mathrm{R}_{\min} < \mathrm{R}_{\mathrm{d}} < \mathrm{R}_{\max},$$

where R_d is the radius of the die top corner (ring), and $R_{\min}$ & $R_{\max}$ are the allowable minimum and maximum values of the radius that depend on the thickness and the type of the material.

3. Implementation examples

In the next Figs. 3 and 4, two implementation case examples are given, showing the automatic design results provided by the KBES for the die, punch and blank holder tools that correspond to the 2 example parts chosen to be shown for the purpose of this paper. Also through these 2 examples, the KBES



Component name: Part to be manufactured

Figure 3a. Part 1 imported

Component name: Die

Figure 3b. Design of the die

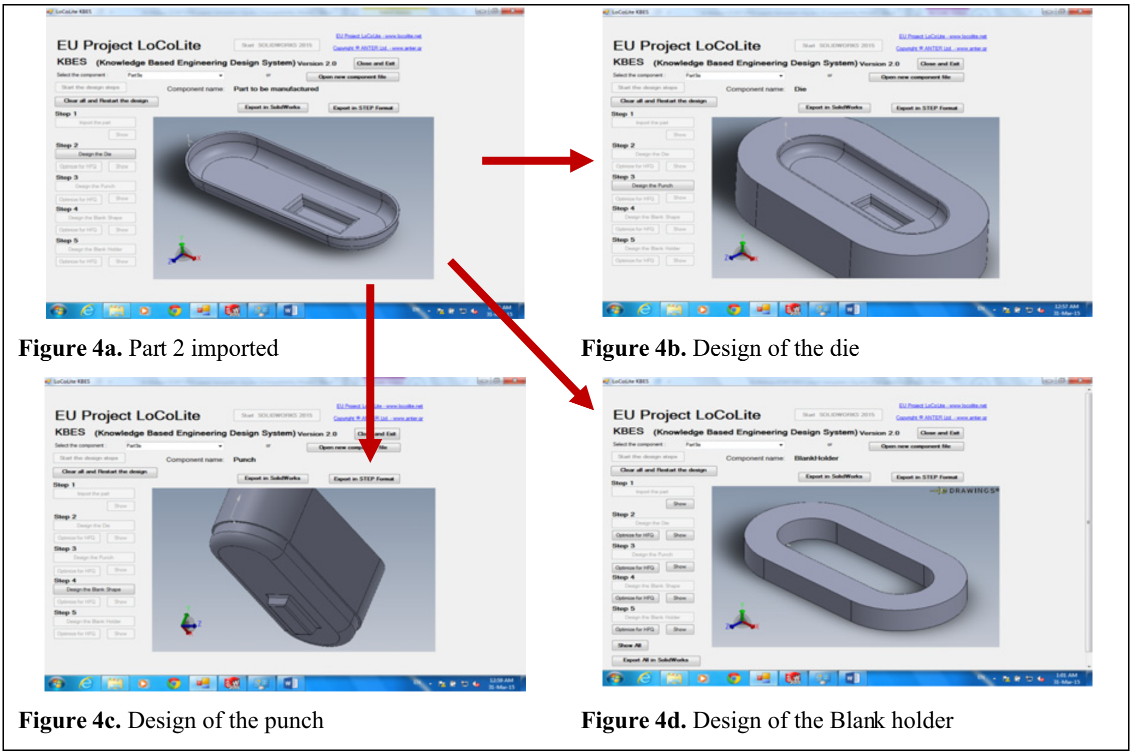
Component name: Punch

Figure 3c. Design of the punch

Component name: BlankHolder

Figure 3d. Design of the Blank holder

Figure 3. KBES design steps & results for case example 1.



Component name: Part to be manufactured

Figure 4a. Part 2 imported

Component name: Die

Figure 4b. Design of the die

Component name: Punch

Figure 4c. Design of the punch

Component name: BlankHolder

Figure 4d. Design of the Blank holder

Figure 4. KBES design steps & results for case example 2.

implementation steps of the design methodology and the main KBES functionalities are in summary presented, starting from the 1st step, which is the import of the 3D model of the part in the system and finishing at the last step, which is the design by the KBES of the blank holder tool. Additionally, during the tools' designs adjustment & improvement stage for the specific HFQ forming process, extra rules were applied for additional changes and refinements of the initial tools geometry concerning e.g.: (a) the contact area of the blank holder in relation with the top surface area of the die and the dimensions of the blank sheet, (b) the thickness of the blank sheet, (c) the size of the die and the punch, in relation with the maximum dimensions (length, width and height) of the part, etc.

4. Conclusions and future work

In this paper an automatic tools design methodology for forming manufacturing processes was presented together with the software system KBES (Knowledge Based Engineering Design System), which was developed for implementing the proposed methodology. This methodology and the software system for the automatic design of the tools, using as main input the part to be manufactured, can be very helpful to tools' designers and manufacturing engineers for improving the efficiency of the whole design & manufacturing process. Additionally, the system gives to the user the possibility to include, update and use extra process-specific knowledge e.g. in the form of rules, that influence, improve and refine the designs of the tools in relation with the specific process under consideration, like in the case of the HFQ process. This makes the proposed methodology and the respective software system suitable, not only for the automatic creation of the basic 3D geometrical models of the tools, but also for process-specific adjustments and refinements of these models.

The KBES software has been developed using Object-Oriented (O-O) and Rule-based methods and programming tools (e.g. NET technologies of Microsoft, C#, Prolog language, etc.) and it is also integrated with the CAD system SOLIDWORKS, through the SOLIDWORKS API interface.

Among others, the future work for the future development of the KBES system will focus on the further gathering and use of additional and more specialized process-specific and experts' knowledge for HFQ within the EU project LoCoLite, which will be useful for further tools design improvements and refinements e.g. in relation with products' shapes with higher degree of complexity and for several additional aluminium material alloys, besides the AA6082 alloy, which is being already used and tested in the LOCOLITE project, etc. Closing we may say that this methodology and the KBES software being developed within the LOCOLITE project, could be useful, besides the specific HFQ forming process for aluminium based products, in other cases of metal forming processes too, using respective process-specific & manufacturing experts' knowledge, for more efficient, successful, fast and low-cost forming tools design.

References

[1] S. Adiga, *Object-oriented Software for manufacturing Systems* (Chapman & Hall, NY, 1993)

[2] J. Barker, G. Palmer, Beginning C# Objects: From Concepts to Code (Apress, CA, 2004)

[3] V. Boljanovic, *Sheet Metal Forming Process and Die Design* (Industrial Press Inc., NY, 2004)

[4] A. Frost, *Introduction to Knowledge Base Systems* (W. Collins & Sons Ltd., UK, 1987)

[5] R. Kerr, *Knowledge-Based Manufacturing Management* (Addison-Wesley, Sydney, 1991)

[6] I. Malpas, *PROLOG: A relational language and its applications* (Prentice-Hall International, New Jersey, 1987)

[7] N. Mekras, N. Karbadakis, P. Kontovazenitis, N. Vaxevanidis, Knowledge Based System Support in Micro-Manufacturing, *Advances in Manufacturing Technology, Proceedings of the 4th International Conference on Manufacturing Research (ICMR 2006)*, Liverpool John Moores University, UK, p. 421–426 (2006)

[8] W. Meyer, *Expert Systems in Factory Management – Knowledge-Based CIM* (Ellis Horwood Ltd., UK, 1990)

[9] C. N. Rowe, *Artificial Intelligence through Prolog* (Prentice-Hall International, Englewood Cliffs, New Jersey, 1988)

[10] H. Schneider, D. Karagiannis, Intelligent Knowledge Bases of CAD Environments: The Hybrid System KANON, *NATO ASI Series*, Springer-Verlag, **F49**, 161–196 (1988)

[11] S. L. Semiatin, *ASM Handbook, Volume 14B, Metalworking: Sheet Forming* (ASM International, Ohio, 2006)

[12] P. Tran, *SOLIDWORKS 2014 Part II – Advanced Techniques* (SDC Publications, USA, 2013)