Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 592–601

Procedia
Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Agent Collaboration in a Multi-Agent-System for Analysis and Optimization of Mechanical Engineering Parts

Paul Christoph Gembarski*

Institute of Product Development,
Leibniz Universität Hannover
An nder Universität 1, 30823 Garbsen, Germany
gembarski@ipeg.uni-hannover.de

Abstract

In mechanical engineering, designers have to review a designed artefact iteratively with different domain experts, e.g. from manufacturing, to avoid later changes and find a robust, optimized design. To support the designer, knowledge-based engineering offers a set of approaches and techniques that formalize and implement engineering knowledge into generic product models or decision support systems. An implementation which satisfies especially the concurrent nature of today's design processes and allow for multi-objective decision-making is multi-agent systems. Such systems consist of entities that are capable of autonomous action, interact intelligently with their environment, communicate and collaborate. In this paper, such a multi-agent system is discussed as extension for a computer-aided design software where the agents take the role of domain experts, like e.g. manufacturing technologists and make suggestions for the optimization of the design of mechanical engineering parts. A focal point is set on the collaboration concept of the single agents. Therefore, the paper proposes the use of an action-item-list as central information and knowledge sharing platform.

Keywords: Multi-Agent System ; Knowledge-Based Engineering ; Computer-Aided Design ; Design Optimization

---

* Corresponding author. Tel.: +49-511-762-5361; fax: +49-511-762-4506.
E-mail address: gembarski@ipeg.uni-hannover.de

10.1016/j.procs.2020.08.061

1. Introduction

The application of software tools in modern product development is state of the art for many disciplines [1]. Such computer aided engineering environments comprise, among others, computer aided design (CAD) tools to define e.g. product shape and production information [2]. Over time, CAD systems for mechanical engineering have developed from tools for 2D line drawing to powerful parametric 3D design systems where a designer is able to modify his parts and assemblies simply by changing values of e.g. dimensions for lengths and adding or deleting features as long as this change can be propagated according to the constraints of the underlying data model [3]. Nowadays, CAD systems are even able to store and process engineering knowledge like dimensioning formulae, design rules and intelligent templates that can easily be adapted to new use cases [4]. This makes knowledge-based engineering available outside from specialized software or expert systems [5].

Two design tasks need to be differentiated: *Routine design* may be entirely automated, like in product configuration [6]. The basis for this is the definition of a product model, which is enriched by defined degrees of freedom and options as well as a process model how the designer or a customer specifies his desired product [7]. So, the CAD model does not only display a single variant but a solution space that comprises all valid variants [8, 9]. In contrast to routine design, *creative design* tasks are characterized by an ill-structured nature, uncertainties and a solution space that is way too large to be modelled, especially considering different physical solution principles in conceptual design [10]. It follows that creative design involves knowledge-based engineering mainly as decision- or problem solving support and co-worker of a human designer [11-13].

1.1. Motivation

Despite the abilities and potential benefits of the above mentioned systems, the development of technical products is still a complex task that usually involves multiple experts from different design domains [10]. According to the idea of the product life cycle, the concurrent engineering paradigm requires product development departments to review design artefacts in every phase of the development process with experts from different functional units like manufacturing (i.e. setup planning, tooling, fixture design), disposal or service engineering [14]. The idea behind is to shorten lead time in development since decision quality is raised and later changes can be avoided [1].

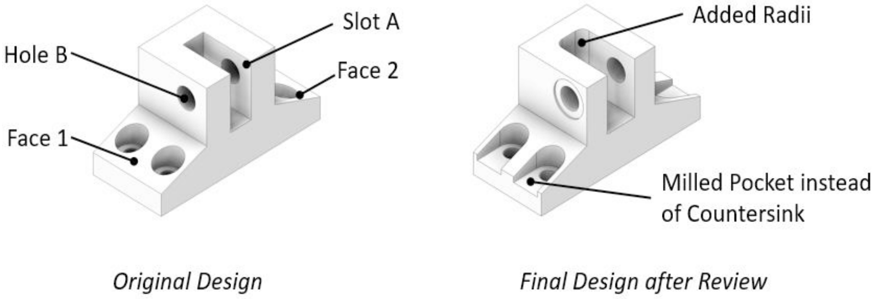


Fig. 1. Mounting Bracket with annotated Faces and Features

E.g. in a design review for the draft of a mechanical engineering part, the geometry can be checked against design guidelines like DfX (Design for eXcellence), where the X is a placeholder for e.g. manufacturability [15]. As an example for the discussions and decisions that occur during the optimization of a subtractively manufactured part stand passages of the design review for the bracket shown in Fig. 1. Regarding the slope of *faces 1* and *2*, the review started with the question whether it has to be milled or is already included in the semi-finished material. The latter is confirmed so that the slope itself is not to be considered as a problem. But with respect to the countersinks of the four drillings, the production planning engineer identified a challenge: Since the surface of the face will be hard and potentially rough, the tool for countersinking might be deflected and is, from his experience, a source for production faults. In order to avoid a tool change, he suggests to use the same tool as for the *slot A* to mill pockets instead of the countersinks since the respecting end mill has a shank that is long enough.

Referring back to CAD and knowledge-based engineering, a system to be used for analysis and optimization of mechanical engineering parts for concurrent engineering needs an implementation that is able to mimic a discussion like above, to capture the expertise of the domain experts and to apply this to any part geometry. Especially the last requirement is challenging since, unlike to product configuration, there is no generic product model where e.g. a known addressable dimension parameter may be restricted by limiting its value range according to a travelling distance of a production machine [16]. More than that, such a system needs the ability to perceive features of the geometry model, their parameters and put them into the context of other knowledge sources like tool lists and manufacturing sequences.

An implementation that satisfies such multi-objective decision-making is possible based on a multi-agent system [17]. These consist of entities that are capable of autonomous action, interact intelligently with their environment, communicate and collaborate [18]. Yet not widespread, their application potential in engineering disciplines is manifold and ranges from control of production systems over modelling and simulation of complex systems like smart grids to workflow management [19]. Moreover, single prototypes aiming at deriving manufacturing sequences based on a CAD model [20] as well as optimizing designs regarding simple design guidelines have been implemented [21].

1.2. Scope and Structure of the Paper

This paper continues this path and presents a concept for such a multi-agent system as extension for a standard CAD software. The agents take the role of domain experts to analyze and make suggestions for the optimization of the design of mechanical engineering parts. The emphasis is put on the organization of the single agents and their collaboration concept. Therefore, the paper proposes the use of an action-item-list as central information and knowledge sharing platform for the single agents.

The paper is organized as follows: Section 2 presents the theoretical background and related work regarding knowledge-based engineering and multi-agent systems related to product development. Afterwards in section 3, the concept for an agent-based analysis and optimization tool is discussed before section 4 presents the organization of the agents working together on an action-item-list. In section 5, a brief introduction in the initial implementation is given showing a feasibility study. The final section 6 presents discussion and conclusion and points out future research potential.

2. Theoretical Background and related Work

2.1. Knowledge-Based Engineering Systems

Knowledge-based engineering (KBE) systems are a subgroup of knowledge-based systems, which have to be understood as a collective term for computer-aided problem-solving tools [22]. In detail, KBE is a set of techniques for (partial) automation of design tasks and to establish product models that are easy to adapt to new requirements [6]. KBE can be seen as evolutionary step in computer aided engineering which is created by the combination of object-oriented programming, artificial intelligence and CAD [23].

From a task-perspective, KBE systems perform different knowledge tasks that deliver artefact descriptions (Fig. 2). *Synthetic design* is designing a system that meets specified requirements. These are first formulated by the user and then operationalized by the KBE system. Hereby hard requirements enable the KBE system to filter possible system designs that have been generated on the basis of knowledge about system creation. Soft requirements enable the system to evaluate and rank multiple valid system designs [24]. *Configuration* means assembling a system out from fully predefined building blocks that are matched via standardized interfaces. From an information science point of view, configuration tasks can be written and solved as constraint satisfaction problem [7]. *Parametrization* aims to eliminate degrees-of-freedom in a variable product model by setting parameter values. Here also apply constraint satisfaction techniques [11].

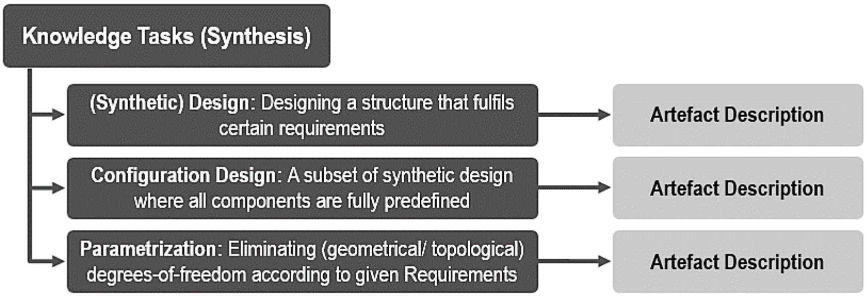


Fig. 2. Knowledge-Based Engineering Tasks that deliver Artefact Descriptions (acc. to [11])

To automatically perform these tasks or combinations of them, a KBE system must have the ability of reasoning [25]. Therefore, a prerequisite is the implementation of two very basic kinds of knowledge [24]: (1) Domain knowledge constitutes a solution space in which a particular solution for a defined set of requirements can be found. It is modelled e.g. by parameter constraints, formulae and design rules for product models [11]. (2) Control knowledge, which is the actual reasoning, states how this solution space is explored and integrates rule-based, model-based or case-based reasoning techniques [26].

Within KBE system implementations, knowledge-based CAD integrates design rules, dimensioning formulae, spreadsheets, macros and interactive applications into the standard modelling tool in engineering design, the CAD system itself [4]. Although available, only single reports about applications exist. Exemplarily stand works from design synthesis in aerospace and automotive engineering [27], conceptual or configuration design in plant engineering [28] or niche design activities like automating fixture design [29]. All of these approaches have in common that knowledge artefacts as well as models have to be implemented explicitly. Regarding the design review mentioned in the motivation above, these traditional approaches are not suitable. A solution space that covers all possible designs e.g. for milled parts is way too large to be integrated into a single KBE system this way.

2.2. Multi-Agent Systems in Product Development

Agent-based modelling and simulation as well as multi-agent systems have been proposed for various applications since the 1980ies [18-20]. The definition of the term *agent* is still subject to standardization but literature widely agrees that an agent is a software entity that operates autonomously without intervention of a human user to complete a task [21]. In order to do so, an agent needs to perceive his environment relevant to the task, to react on changes to the environment and know about consequences [29]. Since a multi-agent system implementation commonly integrates more than one agent, the single agents need the ability to communicate and collaborate with other agents or the user [31].

Such collaboration is essential in multi-agent systems and is expressed in at least two interaction types, i.e. information exchange and negotiation [18]. Regarding the first, different protocols and behaviors were suggested, e.g. from simple message exchange mechanisms with declaration of sender, recipient, intent and action request [32] to more complex interaction control, i.e. defining communicational rights that manage communications in hierarchical agent systems [33]. Examples for negotiation are rule-based and algorithmic auction and bidding models [34, 35], approaches based on classical control and network theory incorporating actuator – sampler – controller – disturbance [36] as well as the implementation of learning strategies and evolutionary approaches like survival-of-the-fittest [37].

Related to product development and engineering design, four core application areas for agent-based systems may be identified from literature:

1. Workflow management in collaborative engineering, e.g. [38, 39]
2. Simulation of system behavior and control circuits, e.g. [19]
3. Synthesis or reengineering of design artefacts, e.g. [30, 31, 40]
4. Analysis of product characteristics according to design guidelines, e.g. [20, 21].

Referring to the design review, the last application area shows similarities since the documented systems are also able to optimize a design. But the majority of approaches focusses just on a narrow set of design guidelines, like e.g. in design for recycling “reduce number of joints”, “reduce number of parts” or “reduce number of different materials in a sub-assembly”. Although this largely corresponds to the expertise of an e.g. eco-design expert, the complexity of these guidelines is low since they can be checked from a bill-of-materials and the CAD-model of the assembly nearly without additional contextual information. For implementation of a multi-agent system with the expertise of production planning and manufacturing specialists, additional knowledge sources need to be addressed and furthermore, the designer himself needs to be simulated in order to judge change requests or defend the original design.

3. Conceptualizing an Agent-based Analysis and Optimization Tool for Mechanical Engineering Parts

Taking the observations from multiple design reviews into account as well as the evaluations from domain experts and related work from literature, a multi-agent system capable of a design review and aiming at the optimization of mechanical engineering parts needs to fulfill following basic requirements:

- The tool must be fully integrated into the designer’s workflow and thus work directly on the CAD files and within the CAD environment. The tool may be integrated in or coupled to, i.e. remote control the CAD system.
- Additionally to geometry, information about tolerances, surface quality and are loads needed to argue machining operations. Some of these may be defined directly on faces of the geometry model in the CAD system, e.g. load paths must be declared by the user.
- For machining operations, a mapping between discrete machines and tool sets must be included in order to reason about setup planning and achievable tolerances since they result from machine-tool-pairs.
- The system must work iteratively, if an adaption is proposed, the new situation has to be analyzed accordingly.
- All conclusions drawn from the system must be explainable and consequences on the dimensions *function fulfillment*, *part mass*, *number of machining operations* and *number of tool changes* must be shown qualitatively.
- The user is active part of the system and can be asked about additional relevant information.

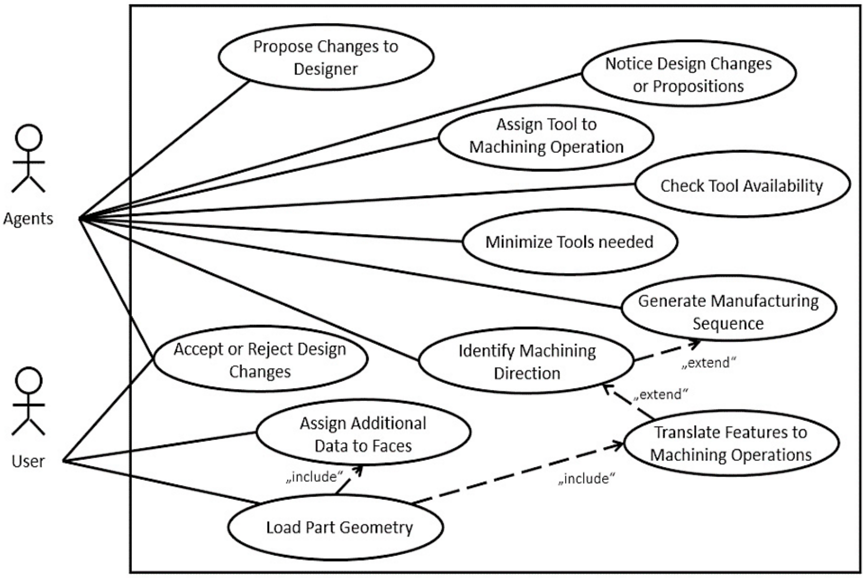


Fig. 3. Use Case Diagram as Specification of Actions in a Design Review (excerpt)

The use case diagram shown in figure 3 displays part of the activities that the system has to perform mimicking a design review. The CAD model is the boundary object for all of these activities. Its components like feature tree, parameter tables and the mathematical representation of the model’s shape itself (list of single faces, enriched with

ID, type, parent feature, bounding box dimensions and technical information like surface quality or tolerances) may be used as a basis for feature recognition and agent perception. Regarding the focus if the paper, such a part repository as well as other knowledge sources like tool databases etc. will not be further discussed here.

4. Organizing for Agent Collaboration

A central point of interest for the system is the sequence of manufacturing operations that are necessary to produce a part since the manufacturing operations are the basis for the analysis of design guidelines. Such a list can be abstracted to an action-item-list, following [33]. Furthermore, the knowledge bases of the agents may be abstracted as production rules since conditions have to be checked and result in further actions. Production rules are clearly readable and single agent rule bases may be split up and organized by a meta-model if it grows too large. Additionally, rules are well-understood in the domain of knowledge-based engineering [41] and Robotic Process Automation [42, 43]. For better understanding, the following description of the control flow and agent interaction will be supported by a running example (figure 4), a summarizing sequence diagram can be found at the end of the section (figure 6).

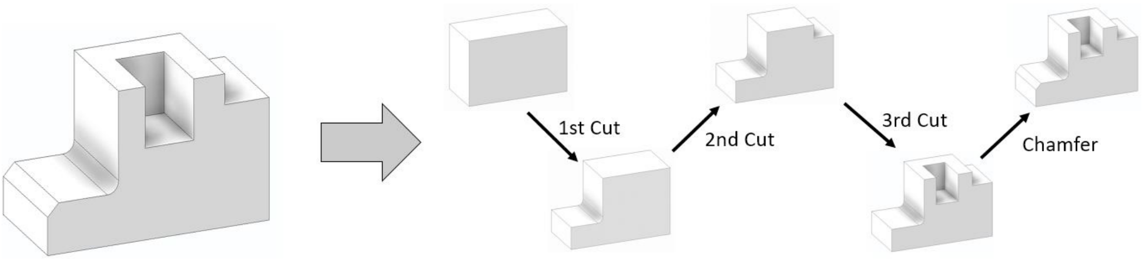
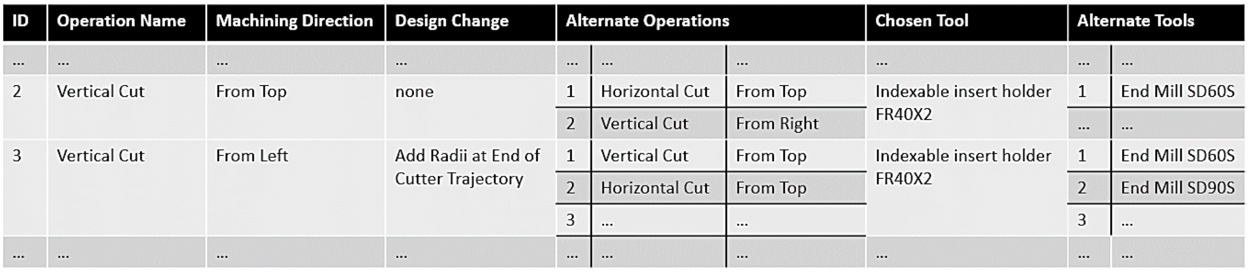


Fig. 4. Running Example

1. When the multi-agent system is initialized on a CAD part model, agents with corresponding feature recognition and perception abilities analyze the model for a first time and create an initial action-item-list (AIL) from the identified manufacturing operations as a table. Referring to the example, a *milling setup agent* identifies three cuts, one of them with inner radius, and a chamfer which have to be manufactured. Each manufacturing operation is characterized by ID, name and machining direction (e.g. from top, from left, 4th axis). The initial machining direction is defined based on the minimum depth. In case of the first cut, machining from the left side results in a smaller depth than from the top.
2. In a second step the *milling setup agent* checks the geometry with respect to the relevant design guidelines. In the example, the third cut is erroneous since it has no tool run-out. The agent rule-base has an entry for such cases which is "*IF milled cut without tool run-out AND milled cut uncontinuous THEN add radii at the end of the cutter trajectory*". This command is stored as design change in the AIL in the corresponding line but not executed in the CAD model. Similar applies for the chamfer which is potentially cosmetic.
3. In the third step, the *milling setup agent* adds alternative machining operations to the AIL as sub-array from a generic reaction pool. This may be manufacturing from different directions or alternate milling operations (e.g. change from end mill to disk milling cutter which leads to another part setup). Each alternative is stored together with a priority which is calculated from the rule-base. Some operations have no alternatives, such as e.g. milling a key slot, these are left blank.
4. As fourth, the *milling tool management agent* establishes a tool database for each entry in the AIL. If there is no corresponding tool available for an operation and the alternative operations sub-array contains no entries, the operation is marked as faulty and an alert is created. If the alternative operations sub-array has entries that contain a related tool set, the first one overwrites the original operation, all other alternatives move up in their priority. All available tools are stored as another sub-array to the manufacturing operations (as well as their alternatives for later access) and prioritized according to the agent rule-base. Figure 5 shows an excerpt from the AIL.



ID	Operation Name	Machining Direction	Design Change	Alternate Operations			Chosen Tool	Alternate Tools	
...	...	...	...	...	...	...	...	...	...
2	Vertical Cut	From Top	none	1	Horizontal Cut	From Top	Indexable insert holder FR40X2	1	End Mill SD60S
				2	Vertical Cut	From Right		...	...
3	Vertical Cut	From Left	Add Radii at End of Cutter Trajectory	1	Vertical Cut	From Top	Indexable insert holder FR40X2	1	End Mill SD60S
				2	Horizontal Cut	From Top		2	End Mill SD90S
				3	...	...		3	...
...	...	...	...	...	...	...	...	...	...

Fig. 5. Action Item List (AIL, excerpt)

5. The *setup planning agent* has the goal to minimize the setup effort. Therefore, he (1) unifies the machining directions by exchanging original and alternative machining operations where possible (in the running example, all machining operations are changed to *from top*) and (2) minimizes the tools used.
6. The *setup planning agents* starts negotiating with the *design agent* about changes to the model that could result in further minimizing setup effort. Related to the running example a question would be if the inner radius of the first cut could be omitted because then the same tool as for the third cut could be applied. If the design agent negates this, the setup planning agent will ask if the same inner radius can be added to the third cut. If the *design agent* agrees then, a corresponding design change is stored.
7. In the seventh step, the *milling setup agent* relaxes further geometric constraints and checks sensitivities. E.g. what would happen if the width of the second cut would be extended or the cut is extended to the full length of the part. If improvements to the setup plan can be found, the according changes are negotiated with the *design agent* and either accepted with a design change stored or rejected.
8. As last steps all design changes are executed and assessed regarding function fulfillment as well as effect on weight, the setup plan is documented and the session protocol is saved.

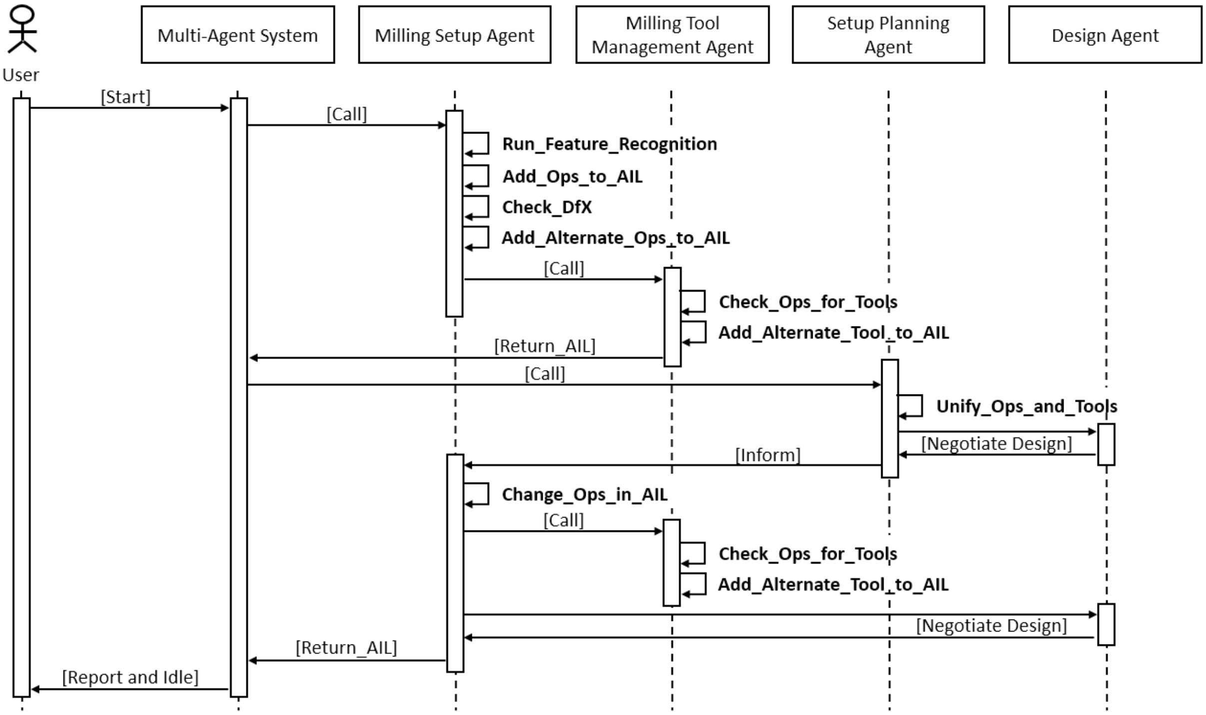


Fig. 6. Sequence Diagram for Agent Interaction on Action Item List

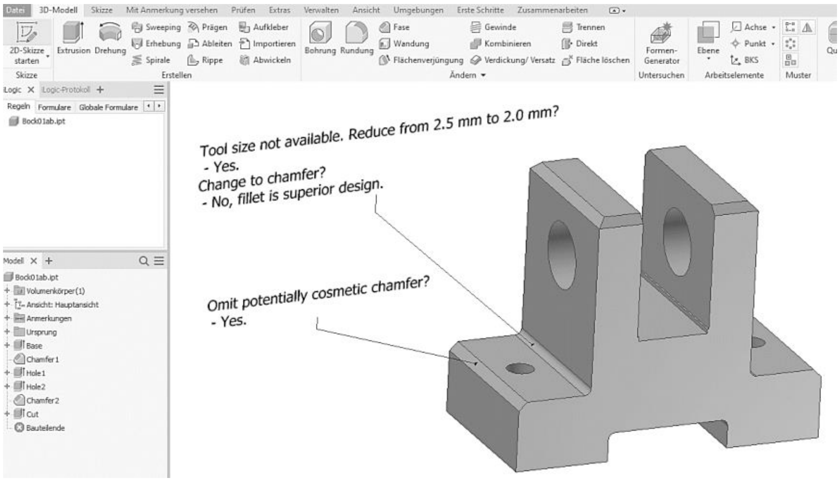
It is worth noting that the call commands between the single agents may be substituted by event triggers. E.g. the *milling tool management agent* may react on the OnChange event of the AIL so that whenever a line is added or changed the tool check is performed.

A second point is the handling of alerts generated during the fourth step of tool assignment. Here also a rule-base is available for the *milling setup agent* who starts negotiating with the *design agent*. The reaction pool contains the change of the size of features, their exchange or creating new features according to given requirements. Referring to the running example, the first cut includes an inner radius. The alert would state that no tool in that size is available. Since omitting the fillet is no option which was already stated in step 6, the *milling setup agent* suggests to reduce the size of the fillet from 5 to 4 mm since such an end mill is existing. The *design agent* might state that 5 mm were meant as minimum due to the stress distribution in that area. Remaining options are then to increase the radius of the fillet, to change the fillet into a chamfer or to change the manufacturing direction to *from front*.

5. Prototype Implementation and Feasibility Study

The prototype system was implemented in the CAD system Autodesk Inventor Professional 2020, using also MS Excel as spreadsheet application and VB.Net as programming interface. For a proof of concept, agents for milling and drilling should be implemented since both operations rely on design guidelines which can be implemented with acceptable effort but which go beyond a simple assessment of the feature tree or the bill of materials. In order to reduce programming effort, the execution of design changes was excluded from the prototype implementation. Instead, the system adds comments on geometry elements which the system would like to be changed. The comments then show the negotiation of the agents in plain text.

As test case, a clevis was chosen which had some similar features compared to the bracket mentioned in the introductory part of this paper (figure 1). After the part file had been loaded to the CAD system, the analysis and optimization tool was started.



Tool size not available. Reduce from 2.5 mm to 2.0 mm?
- Yes.
Change to chamfer?
- No, fillet is superior design.

Omit potentially cosmetic chamfer?
- Yes.

Fig. 6. Inventor Part Document with annotated Agent Dialogs

The system successfully identified the four cuts and three drillings (the clevis bore is considered as one) in the model. Regarding the outer cuts, the machining direction was defined from top in order to minimize setups and taking into account that an indexable insert holder with long shank is available. The agents identified the chamfer as cosmetic and suggested to omit this. Since no tool with inner radius of 2.5 mm was included in the tool database, the agents reduced the size of the radius. Afterwards, according to steps six and seven, the agents negotiated about replacing the fillet with a chamfer since this could be manufactured with the same tool as the top chamfers of the clevis. Since the fillet is the superior design due to stress reasons, no design change was prompted.

6. Discussion and Conclusion

The implemented prototype showed the action and functional principles of the designed agent-based analysis and optimization tool. It has been verified that the action-item-list provides a basic mechanism for the agents' organization and the information exchange. In further implementations this will be extended towards automatically changing the CAD model where the next challenge is to safely identify the model elements, e.g. faces or edges of the geometry, where such a change has to be referred to.

It has to be noted that each manufacturing technology implies particular expertise and involves own design guidelines that partly conflict with other ones. In principle, a mechanism for conflict management could be provided by rules, like exemplarily presented in the negotiation between setup and design agent. Nonetheless, the decision which is the superior design then forces the integration of knowledge about the functions of the modelled part, their needs for features and accuracy as well as the requirements from neighbor parts in the assembly. In this context, it should be taken into account that the involvement of a human designer for judging alternative designs which were proposed by the multi-agent system, is also a considerable interaction format.

A limitation of the implemented prototype system, which may also be starting point for further research, is its focus on prismatic geometries which makes the manufacturability check comparatively easy. Geometries that allow free-form surfaces like in casting or in additive manufacturing require a different level of perception. In particular cases, e.g. the casting design guideline *avoid material accumulations*, the use of an artificial intelligence for image or voxel processing could be a valuable approach.

Another limitation is the explicit knowledge modelling. All agent rule-bases are prescribed, at this stage the agents do not modify them by themselves. Here, it would be beneficial to integrate the ability of learning in order to follow the principles of intelligent agents. Starting points in this matter might be the integration of a case-based reasoning system where the action-item-list serves in correspondence with the setup list and part geometry as case representation.

References

[1] Vajna, Sandor (2014) "Integrated Design Engineering." Berlin Heidelberg, Springer

[2] Stroud, Ian, and Hildegarde Nagy. (2011) "Solid modelling and CAD systems." Berlin Heidelberg, Springer

[3] Shah, Jami J. (2001) "Designing with parametric cad: Classification and comparison of construction techniques." *Geometric Modelling*: 53–68.

[4] Hirz, Mario, Dietrich, Wilhelm, Gfrerrer, Anton, and Johann Lang (2013) „Integrated computer-aided design in automotive development." Berlin Heidelberg, Springer

[5] Hopgood, Adrian .A. (2016) "Intelligent systems for engineers and scientists." Boca Raton, CRC press

[6] Verhagen, Wim .J.C., Bermell-Garcia, Pablo, van Dijk, Reinier E.C., and Richard Curran (2012) "A critical review of Knowledge-Based Engineering: An identification of research challenges." *Advanced Engineering Informatics* **26**(**1**): 5-15.

[7] Felfernig, Alexander, Hotz, Lothar, Bagley, Claire, and Juha Tiihonen (2014) "Knowledge-based configuration: From research to business cases." Burlington, Morgan Kaufmann

[8] Gembarski, Paul Christoph, and Roland Lachmayer (2018) "Solution space development: conceptual reflections and development of the parameter space matrix as planning tool for geometry-based solution spaces." *International Journal of Industrial Engineering and Management* **9**(**4**):177-186.

[9] Skarka, Wojciech (2007) "Application of MOKA methodology in generative model creation using CATIA." *Engineering Applications of Artificial Intelligence* **20**(**5**): 677-690.

[10] Pahl, Gerhard, Beitz, Wolfgang, Feldhusen, Jörg and Karl-Heinz Grote (2007) "Engineering Design - A Systematic Approach." Berlin Heidelberg, Springer

[11] Gembarski, Paul Christoph, Bibani, Mehdi, and Roland Lachmayer (2016) "Design Catalogues: Knowledge Repositories for Knowledge-Based-Engineering Applications.", in Marjanović, Dorian, Štorga, Mario, Pavković, Neven, Bojčetić, Nenad and Stanko Škec (eds) *Proceedings of the DESIGN 2016 14th International Design Conference*, Glasgow, The Design Society

[12] Li, Haibing, Gembarski, Paul Christoph, and Roland Lachmayer (2018) "Template-Based Design for Design Co-Creation.", in Wodehouse, Andrew, Georgiev, Georgi, Cascini, Gaetano, Dekoninck, Elies, and Chris Snider (eds) *Proceedings of the 5th International Conference on Design Creativity (ICDC2018)*, Glasgow, The Design Society

[13] Brockmöller, Tim, Gembarski, Paul Christoph, Mozgova, Iryna, and Roland Lachmayer (2017) "Design catalogue in a CAE environment for the illustration of tailored forming.", in *Proceedings of the 59th Ilmenau Scientific Colloquium*, Ilmenau, TU Ilmenau

[14] Ullman, David G. (2009) "The mechanical design process." New York, McGraw Hill Higher Education

[15] Huang, George Q. (1996) “Design for X - Concurrent engineering imperatives”, Berlin Heidelberg, Springer

[16] Gembarski, Paul Christoph, Sauthoff, Bastian, Brockmöller, Tim, and Roland Lachmayer (2016) “Operationalization of Manufacturing Restrictions for CAD and KBE-Systems.” in Marjanović, Dorian, Štorga, Mario, Pavković, Neven, Bojčetić, Nenad and Stanko Škec (eds) *Proceedings of the DESIGN 2016 14th International Design Conference*, Glasgow, The Design Society

[17] Mönch, Lars, and Marcel Stehli (2006) “ManufAg: a multi-agent-system framework for production control of complex manufacturing systems.” *Information Systems and e-Business Management* **4**(**2**): 159-185.

[18] Jennings, Nicholas R. and Michael J. Wooldridge (1998) “Agent Technology - Foundations, Applications, and Markets.” Berlin Heidelberg, Springer

[19] Taylor, Simon (2014) “Agent-based Modeling and Simulation.” Basingstoke, Palgrave Macmillan

[20] Sun, J., Zhang, Y.F., and A.Y.C. Nee (2001) “A distributed multi-agent environment for product design and manufacturing planning.” *International Journal of Production Research* **39**(**4**): 625-645.

[21] Dostatni, Ewa, Diakun, Jacek, Grajewski, Damian, Wichniarek, Radoslaw, and Anna Karwasz (2016) “Multi-agent system to support decision-making process in design for recycling.” *Soft Computing*, **20**(**11**): 4347-4361.

[22] Milton, Nick.R. (2008) “Knowledge technologies.” Monza, Polimetrica sas.

[23] Chapman, Craig B., and Martyn Pinfold (2001) “The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure.” *Advances in Engineering Software* **32**(**12**): 903-912.

[24] Schreiber, Guus, Wielinga, Bob, de Hoog, Robert, Akkermans, Hans and Walter van de Velde (1994) “CommonKADS: A comprehensive methodology for KBS development.” *IEEE expert* **9**(**6**): 28-37.

[25] Sabin, Daniel, and Rainer Weigel (1998) “Product configuration frameworks - a survey.” *IEEE intelligent systems* **13**(**4**): 42-49.

[26] Gembarski, Paul Christoph (2019) „Komplexitätsmanagement mittels wissensbasiertem CAD – Ein Ansatz zum unternehmenstypologischen Management konstruktiver Lösungsräume.“ Garbsen, TEWISS.

[27] La Rocca, Gianfranco (2012) “Knowledge based engineering - Between AI and CAD. Review of a language based technology to support engineering design.” *Advanced engineering informatics* **26**(**2**), 159-179

[28] Hvam, Lars, Mortensen, Niels H., and Jesper Riis (2008) “Product customization.” Berlin Heidelberg, Springer.

[29] Boyle, Iian, Rong, Yiming, and David C. Brown (2011) “A review and analysis of current computer-aided fixture design approaches.” *Robotics and Computer-Integrated Manufacturing* **27**(**1**), 1-12.

[30] Fougères, Alain-.Jerome, and Egon Ostrosi (2018) “Intelligent agents for feature modelling in computer aided design.” *Journal of Computational Design and Engineering* **5**(**1**), 19-40.

[31] Chen, Yong, Liu, Ze-Lin, and You-Bai Xie (2014) “A multi-agent-based approach for conceptual design synthesis of multi-disciplinary systems.” *International Journal of Production Research* **52**(**6**), 1681-1694.

[32] Hmida, Faten Ben, Anne Seguy, and Rémy Dupas (2012) “MultiAgent Systems for Production Planning and Control in Supply Chains.” In: Omatu, S. et al. (eds.) Distributed Computing and Artificial Intelligence, pp 205-212.

[33] Mönch, Lars, and Marcel Stehli (2006) “ManufAg: a multi-agent-system framework for production control of complex manufacturing systems.” *Information Systems and e-Business Management* **4**(**2**), 159-185.

[34] Ganzha, Maria, and Marcin Paprzycki (2007) “Implementing rule-based automated price negotiation in an agent system.” *Journal of Universal Computer Science* **13**(**2**), 244-266.

[35] Adhau, Sunil, Murali Lal Mittal, and Abhinav Mittal (2012) “A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach.” *Engineering Applications of Artificial Intelligence* **25**(**8**), 1738-1751.

[36] Ding, Lei, Qing-Long Han, and Ge Guo (2013) “Network-based leader-following consensus for distributed multi-agent systems.” *Automatica* **49**(**7**), 2281-2286.

[37] Banjo, Olatunde, Stewart, David, and Maria Fasli “Optimising Mixed Model Assembly Lines for Mass Customisation: A Multi Agent Systems Approach.” In: Advances in Manufacturing Technology XXX: Proceedings of the 14th International Conference on Manufacturing Research, pp. 337-343 (2016).

[38] Huang, Ching-Jen., Trappey, Amy JC., and Yin-Ho Yao (2006) “Developing an agent-based workflow management system for collaborative product design.” *Industrial Management & Data Systems* **106**(**5**), 680-699.

[39] Juan, Yeh-Chun., Ou-Yang, CHao, and Jiun-Shiung Lin (2009) “A process-oriented multi-agent system development approach to support the cooperation-activities of concurrent new product development.” *Computers & Industrial Engineering* **57**(**4**), 1363-1376.

[40] Huang, Chun-Che (2004) “A multi-agent approach to collaborative design of modular products.” *Concurrent Engineering* **12**(**1**), 39-47.

[41] Gembarski, Paul Christoph, and Roland Lachmayer (2017) “How Rule-based Systems impact Product Complexity.” *Annals of the Faculty of Engineering Hunedoara-International Journal of Engineering* **15**(**1**), 17-24.

[42] Davenport, Thomas H. (2018) “The AI Advantage: How to Put the Artificial Intelligence Revolution to Work.” Cambridge, MIT Press

[43] Aguirre, Santiago, and Alejandro Rodriguez (2017) “Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study.” In: Figueroa-Garcí, Juan Carlos, López-Santana, Eduyn Ramiro, Villa-Ramíre, José Luis, and Roberto Ferro-Escobar (eds) *Applied Computer Sciences in Engineering. WEA 2017. Communications in Computer and Information Science, vol 742*. Berlin Heidelberg, Springer