Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 207 (2022) 946–955

Procedia Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Development of a knowledge-based and collaborative engineering design agent

Stefan Plappert[a,*], Paul Christoph Gembarski[a], Roland Lachmayer[a]

[a]*Institute of Product Development, Leibniz University of Hannover, An der Universität 1, 30823 Garbsen, Germany*

Abstract

In order to avoid errors in engineering design that affect the later product life cycle, especially the manufacturing process, an analysis or evaluation has to be performed at the earliest possible stage. As this evaluation is very knowledge-intensive and often this knowledge is not directly available to the engineer, this paper presents an approach for a knowledge-based and collaborative engineering design agent. The technology based on multi-agent systems enables problem-solving support by an autonomous knowledge-based system which has its own beliefs, goals, and intentions. The presented approach is embedded in a CAD development environment and validated on an application example from engineering design.

Keywords: Engineering Design Agent; Multi-Agent Systems; Knowledge-based Systems; Design Evaluation; Decision-Support Systems

1. Motivation

Imagine the following situation: an engineer designs a possible solution for a part that he or she thinks is the "best" solution and passes it on to the manufacturing engineer. He or she evaluates the design for possible manufacturability and determines that the design is not manufacturable without modifications. The necessary modifications can lead to changes and efforts, each with different requirements, at various stages of the development process. E.g., an adaptation of the design can lead to changes in the functionality of the part, for which the design engineer has to make the decisions and execute further actions. Other possibilities would be the purchase of new tooling or the production of a new fixture to ensure manufacturability, for which an estimation of the manufacturing engineer is necessary, as he or she has the necessary knowledge of the manufacturing processes and tooling. This conflict can only be solved if, first, the manufacturing engineer and the designer communicate with each other to make their positions clear and, second, a coordinated solution has to be found so that the overall effort for the modification can be reduced, regardless of the location at which effort is caused. The resolution of the conflict is difficult in practice because there is a lack of regular

* Corresponding author. Tel.: +49-511-762-14989.
E-mail address: plappert@ipeg.uni-hannover.de

10.1016/j.procs.2022.09.150

communication between the design and manufacturing departments, also due to a geographical distance between the departments, so the design department does not know which tools are available in manufacturing, causing unnecessary iterations, which in turn increase the duration of the design process and the costs [5, 7, 8].

To overcome this conflict, agent-based technology provides a natural means of modeling a problem based on the paradigm of distributed artificial intelligence, since real-world entities and their interactions can be directly represented by autonomous problem-solving agents [16]. For this purpose, this paper will present the development of a knowledge-based and collaborative design agent, which represents the knowledge of the involved experts and can interact with other agents and the user through communication capabilities. Thus, the designer should be able to estimate the manufacturability of his design already during the development process, as the knowledge about the existing manufacturing processes is stored. The assistance system can be applied from two perspectives: the analytical application, in which the design is checked for design guidelines, standards, and manufacturing processes, and the synthetic application, in which, for example, an 80 % design is automatically completed for production in the CAD program.

This paper is organized as follows: Section 2 presents the theoretical background and related work on knowledge-based and multi-agent systems and defines the research questions. The methodological procedure is explained in Section 3. Then, Section 4 introduces the specification, architecture, and implementation of the knowledge-based and collaborative engineering design agent. The application of the engineering design agent approach is applied in Section 5 with an application example and discussed and summarized in Section 6.

2. Theoretical Background and Related Work

2.1. Knowledge-based Systems

One possibility to represent specific domain knowledge are knowledge-based systems (KBS), which use digitally processed expert knowledge and simulate the actions of an expert in going through a solution process [25]. The structure of a KBS is divided into the main components of knowledge base, which serves as a database for the system, inference engine, which makes inferences based on the expert knowledge, and interface to the environment, which enables the KBS to be used by the user [13].

There are two possible approaches for the representation of knowledge within a knowledge-based system, one is the direct and explicit modeling of knowledge in the form of rules, constraints, and case bases and the other is the indirect and implicit modeling of the boundary conditions and restrictions, so that an optimization algorithm can be used to explore the solution space. The explicit form of knowledge representation is increasingly used in the design phase of the product development process, especially in knowledge-based CAD [30], e.g., in expert systems that were originally designed for product configuration and design automation [14]. The implicit form of knowledge representation as performed in engineering design optimization (EDO) often requires knowledge about the stage of design, design variables and their minimum and maximum limits (independent variables), constraints, measurement of the design performance (dependent variables), design parameters and relationships between the independent and dependent variables, i.e., a design evaluation model [33]. Due to the uncertainty in real-world problems it is useful to consider the uncertainty in the modeling, e.g., by Bayesian decision networks and to search for robust solutions in the optimization, which are insensitive to uncontrolled factors [10].

2.2. Agents and Multi-Agent Systems

An intelligent agent consists of autonomous knowledge-based systems that have their own resources and expertise and can interact with others by perceiving, reasoning, adapting, learning, cooperating, and delegating in a dynamic environment [21]. Autonomous action is one of the main characteristics that distinguish intelligent agents from conventional software [6]. In multi-agent systems (MAS), the common goals are achieved through collaboration since no single agent has all the knowledge required to solve a problem [27]. MAS are an effective approach for modeling and designing systems involving humans due to their autonomy and social nature [40].

In related literature on MAS in product development, several approaches exist that support the designer in evaluating his or her design and include further aspects of the product life cycle for this purpose [31]. The focus varies between human-centered approaches, which focus on the support for the designer, and process-centered approaches, which aim at a fluent manufacturing process.

Kratzer et al. [20] developed an agent-based design support system called ProKon, which proactively reviews a CAD model while the engineer is working on it. Here, the agents review the CAD model with respect to objects, such as machine elements, and rules. Also, Dostatni et al. [6] present an agent-based system in a CAD environment that helps the designer to select the material considering its ecological properties and its compatibility with other materials, as well as the type and number of connections. Madhusudan [22], on the other hand, supports the designer at the assembly level through agent-based process coordination to support the use of domain-specific product and process knowledge to enable flexible process design and execution.

Concerning the manufacturability of components, Sun et al. [37] developed a system that uses agents to identify the features of a Unigraphics model and to evaluate, e.g., the tool accessibility. This system has been extended firstly by Jia et al. [17] in terms of a modular architecture so that the agents can be easily reused and updated and secondly by Mahesh et al. [23] who use WebMAS to allow the agents to communicate over the Internet and add the aspects of process planning, scheduling and production monitoring to the system. Medani and Ratchev [24] present a prototype web-enabled system based on a STEP AP224-compliant product data model for rapid product manufacturability assessment in the extended enterprise using collaborative autonomous design and manufacturing agents. Feng [8] starts earlier in the product development process as intelligent agents are used to support preliminary design and integration with other engineering activities in the product development process by having the agents access design specifications of processes and equipment for manufacturing different types of parts, which are represented in a knowledge base.

Based on the related works on KBS and MAS, it can be stated that usually MAS are used for the analytical application, which is often limited to the analysis of CAD models and the subsequent recommendation of hints and suggestions, and the synthetic application is often reserved for KBS, which perform an adaptation of the CAD model but operate for this purpose in a closed solution space. Only Kratzer et al. [20] perform a manipulation of the CAD model in its MAS, but the adaptation is performed in a defined solution space and thus is more equivalent to a configuration. Since there is so far no approach for the combination of KBS and MAS, which allows an automated adaptation of the CAD model even in open solution spaces, an interaction with the designer, and a decentralized capture, storage, and use of knowledge, this paper presents the development of a knowledge-based and collaborative engineering design agent. For this purpose, the following research questions have been identified:

- What functions does a knowledge-based and collaborative design agent need to contain to intelligently support the designer in the evaluation of his or her design and its automated adaptation?
- How can a methodical approach be designed to adapt this design agent to the specific tasks of engineering design and embed it in a human-centered multi-agent system?

3. Methodological Procedure

For the research design in the development of the knowledge-based and collaborative design agent, the specific characteristics and requirements of MAS and KBS, as well as their combination, have to be addressed, since communication is essential for problem-solving in MAS and the formalization of expert knowledge is a central aspect in KBS. In addition, the area of application and the task of the design agent has to be taken into account, as the agent is supposed to be embedded directly into a CAD development environment and support the designer in the evaluation of the engineering design. Due to this context, the CommonKADS methodology (*Common Knowledge Acquisition and Documentation Structuring Methodology*) [34] was chosen, because it captures the key properties of the system and its environment by building separate models. This method was originally developed for KBS development before its use was also proposed for MAS development [15].

The CommonKADS methodology consists of six models that represent different levels of abstraction within the development [4, 19]. The *organization model* identifies and analyzes specific organizational and application contexts into which the MAS will be implemented [15]. The *task model* describes the tasks involved in realizing a function in the organization, independent of an agent performing the task [4]. The *agent model* describes all relevant capabilities and characteristics of agents that can perform the tasks identified in the task model, which may be a human, computer software, or other entity [15]. A central model in the CommonKADS methodology is the *expertise model*, which models an agent's problem-solving behavior in terms of the knowledge applied to perform a particular task [34]. Since there are usually several agents involved in performing the tasks described in the task model, it is important to model the communication between the agents, this is done in the *communication model* [34]. While the other five models deal with analysis and conceptual design, the *design model* is used to describe the architecture and technical design of the MAS in order to implement it [15].

Since the design model represents an extensive aspect of the development alone, the formalization of knowledge according to the MOKA methodology (Methodology and tools Oriented to Knowledge-based Engineering Applications) [36] is additionally used for representation of the domain knowledge and connection to the CAD system. This is because the MOKA methodology provides a comparatively wide range of applications in the use of CAx systems for the application areas specialized in engineering knowledge [35]. Especially the phases *package* and *activate* are interesting for the application in the design model, because in the package phase the domain knowledge is implemented after the formalization and in the activate phase the system is distributed, installed and used [36].

4. Knowledge-based and Collaborative Engineering Design Agent

4.1. Specification

Decisions have to be made by the designer during the entire product development process, which can be divided into *task clarification*, *concept*, *embodiment design*, and *detailed design* [7]. In form design, a part of embodiment design in which the form and material properties are directly specified, the designer has to decide how the requirements for a product can be fulfilled as effectively as possible by the product properties [39]. In addition, indirectly specified product properties for the product life cycle, e.g., material or production costs, and conditions between the form design elements have also to be taken into account. To avoid errors within embodiment design and in the later product life cycle, analysis or evaluation must be performed, preferably in an interdisciplinary team, to use the different experiences of the team members for a comprehensive assessment [39]. Design guidelines such as DfX (Design for eXcellence) can also be used here, for example, to check the geometry for its manufacturability (design for manufacturing) and assemblability (design for assembly) [38]. As the designer works in a highly networked environment and the modeling procedure constantly alternates between synthesis and analysis activities (creative-corrective procedure) [7], a tool that supports the designer in evaluating his or her design should be directly integrated into this workflow. Based on this assignment to the product development process, the following requirements for an assistance system for design evaluation can be derived:

- The tool should be fully integrated into the designer's workflow and thus work directly with the CAD files and the information they contain [9].
- The tool should represent the knowledge of different experts from cross-functional disciplines so that a holistic evaluation of the design can be performed.
- The tool should support negotiation in collaborative design problems so that information can be shared, knowledge of other experts' perspectives and intentions can be gained, and changes can be made based on the shared information [18].
- The tool should assist in documenting the reasoning process, the possible alternatives, and the justification for the final decision made.
- The designer should be involved in the evaluation process (human-in-the-loop), especially for activities that are easy for humans but very difficult to represent in a system.

MAS can be used to represent these requirements in a system because each agent represents a different expert from various disciplines, the agents can communicate and negotiate with each other, and the user can be involved in the decision-making process. In addition, the MAS has interfaces to the environment, e.g., a CAD system, and can document the decisions made and their alternatives. The *organization model* in which the MAS is to be used in engineering design, in particular in form design, can be derived from the context described. Furthermore, the *task model* represents an assistance system that evaluates a CAD model with regard to its manufacturability.

4.2. Architecture of the Engineering Design Agent

In order to build a MAS as described in the specification (section 4.1), it is useful to design agents that have standardized interfaces, so that a flexible combination of agents is possible. Nwana et al. [26] describe with their *ZEUS Generic Agent* a structure that contains the general functionalities of an agent, such as the handling of messages, the execution of actions, the use of information from databases, as well as the coordination and planning of tasks. However, the provision of these functionalities for the requirements of the engineering design domain, such as the processing of geometry information and the comparison with standards, is missing, so the use of MAS in the engineering domain can only be implemented with great effort. The goal of the development of a knowledge-based and collaborative engineering design agent is to represent the basic functionalities based on validated approaches and to extend them by the requirements from the specification so that a template for the designers can be provided and they can concentrate on the application for the concrete use case in engineering design.

For communication within the *communication model*, a FIPA-compliant solution is sought, which has defined a set of interaction protocols based on communicative acts. Therefore, the primary features of the FIPA ACL are used to employ different content languages and manage conversations through predefined interaction protocols, such as the contract-net protocol for establishing agreements and several types of auctions [1]. This can either enable collaborations between agents that work together towards a specific goal or coordinate tasks between agents so that there is a division of work and each agent can provide its specific capabilities to the community.

Problem-solving behavior, which is a central aspect of the *expertise model*, is represented in the case of the engineering design agent by a belief-desire-intention (BDI) architecture [11], which is probably the best known and most popular deliberative agent architecture [1]. The deliberative agent architecture allows the agent to pursue explicit goals that are constantly compared with current beliefs to select appropriate plans for implementation and provide them as intentions for the agent to execute. This architecture additionally enables the agent to negotiate effectively, since agents can also take into account beliefs, desires, and intentions of other agents [27]. Furthermore, by using ontologies, which can be viewed as a knowledge representation of parts of the world, captured entities, ideas, and events, with their properties and relationships [12], knowledge about the environment can be shared with the agent. Due to the specific field of application of the agent in engineering design, provisions for a common domain understanding and the management of the ontology can be established, e.g., to store characteristics of mechanical components and their manufacturing properties.

The *agent model* in figure 1 describes all relevant capabilities and properties of the engineering design agent, as well as the embedding of the agent in the design environment. The interface to the CAD system enables the agents to directly support the designer in the design of mechanical products. Due to the standardized communication via a communication server, decentralized agents or additional human experts can also be integrated. Standardized queries can be used to access data and information that is already stored in databases, such as existing tools in the manufacturing machines. To ensure collaboration between the agents, they need to have a common understanding of the task on the one hand (ontology database) and on the other hand, they need to know the capabilities of the other agents (acquaintance database). The engineering design agent can be divided into two levels a *global* and *local* one, as on the one hand the communication and collaboration to other agents within the MAS cover global aspects and on the other hand, the perception, action, and reasoning are done on a local level, as these activities are performed by the agent independently. To pick up the knowledge-based nature of the engineering design agent, the agent can further be divided into an *inference engine*, a *knowledge base*, and an *interface to the environment*. A detailed description of the individual modules and the implementation in a development environment is given in the following section 4.3.

4.3. Implementation as Template

For the implementation within the *design model*, a MAS platform called SPADE 3.0 (Smart Python Agent Development Environment) developed by Palanca et al. [29] is used. SPADE was chosen for several reasons; on the one hand, SPADE enables FIPA-compliant communication using FIPA performatives, and on the other hand, SPADE uses a modern and full-featured programming language such as Python, which is one of the most widely used programming languages for artificial intelligence (AI) applications [3, 29]. This also addresses criticism of JADE, as the choice of Java as the only programming language for designing JADE agents is considered inappropriate in many situations (such as agent-oriented programming) [2]. Another unique feature of SPADE is its communication over

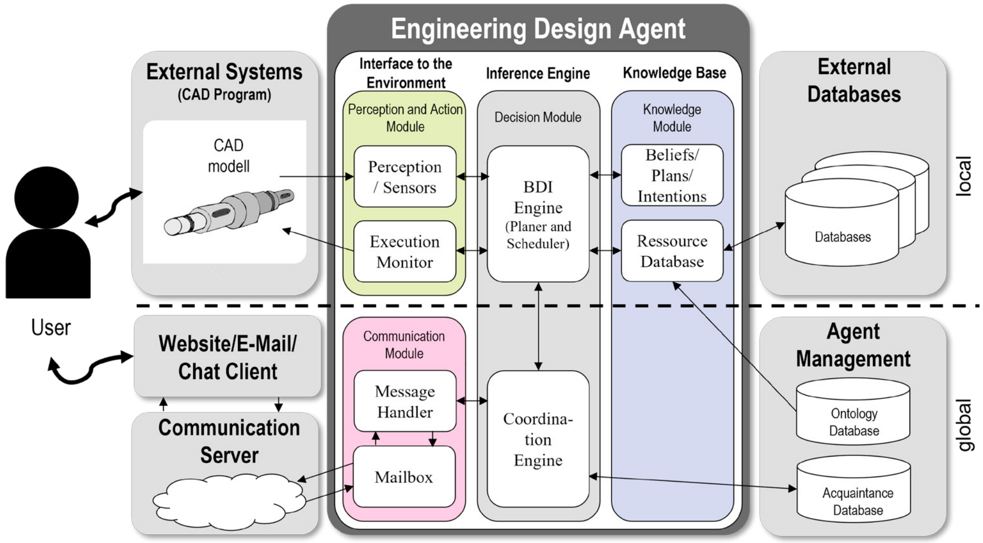

Fig. 1. Knowledge-based and Collaborative Engineering Design Agent.

XMPP (Extensible Messaging and Presence Protocol), which is an open protocol for instant messaging and presence notifications, allowing for natural human-in-the-loop integration [29]. The core of the infrastructure for the engineering design agent is the Python-based SPADE platform, which is extended by communication via XMPP server based on Prosody IM, access to relational databases like SQL, and an interface to the API of the CAD program Autodesk Inventor Professional.

The engineering design agent is divided into four modules *communication module*, *knowledge module*, *decision module* and *perception and action module*, so that the most important functionalities can be developed and adapted separately from each other. The *communication module* ensures communication to other agents, the user, and other human experts by sending messages and in turn receiving and evaluating them. The *knowledge module* contains the agent's knowledge and beliefs so that the agent can manage its beliefs, plans for action for specific events, and intentions for changing the environment. In addition, the agent either accesses knowledge from external databases or uses stored knowledge in the form of rules, constraints, and case bases. The *perception and action module* enables the agent to perceive and influence its environment or external systems. In the case of the engineering design agent, information from the CAD model is read out and adjusted after a reasoning process is completed. The *decision module* represents the heart of the engineering design agent, which decides how to handle requests from other agents or coordinate tasks on its own and makes conclusions based on its beliefs, and executes them according to stored plans. In the following the components of the engineering design agent according to figure 1 are described in detail:

- The *Mailbox* manages communication between the agent and other agents by sending and receiving messages via xmpp or emails.
- The *Message Handler* handles incoming and outgoing messages by reading or adding the meta-data such as FIPA ACL parameters and prepares the messages for further processing by other components of the agent.
- The *Acquaintance Database* registers the agents in the MAS in a relational database and manages their services similar to a yellow pages service, so that the coordination engine can use the information contained to make cooperation agreements with other agents.
- The *Ontology Database* contains ontologies and classifications of features and their relation to manufacturing processes so that an assignment to the capabilities of agents is possible and an evaluation and assessment of components is supported. Linking ontologies together can be supported by the use of graph databases.
- The *Resource Database* contains explicit knowledge of the agent in the form of rules, constraints, and case bases, such as standards or design guidelines, and enables the agent to access external resources, such as existing manufacturing machinery and tools.

- The *Beliefs, Plans and Intentions* are more dynamic compared to the resource database and contain the agent's current beliefs, stored plans, and intentions and are constantly reviewed and updated as the agent uses them.
- The *Perception and Sensors* perceive the environment or external systems, e.g., CAD models within the CAD program Autodesk Inventor are analyzed using feature recognition, where areas from faces of the BRep (Boundary Representation) model are identified which combine a common function or manufacturing step.
- The *Execution Monitor* makes changes in the environment or external systems, e.g., to a CAD model by either adding or removing features or adjusting parameters, so that correction of the CAD model can be automated.
- The *Coordination Engine* represents the agent externally to other agents or the user, so it is also responsible for coordinating interactions as initiator or participant, using well-known FIPA coordination protocols and strategies, such as various auction protocols or the contract-net protocol.
- The *BDI Engine* represents the control center of the agent, the BDI architecture ensures the autonomous action capability of the agent by constantly updating the beliefs and goals of the agent and comparing them with the environment. In case of perceived or communicated changes, the agent can react quickly by checking possible actions against plans and storing them as possible intentions for execution or passing them on to the execution monitor. By using the AgentSpeak programming language, standardized BDI properties can be programmed and integrated into the SPADE environment.

The presented framework for the engineering design agent represents a template, which already has the interfaces to the design environment of the designer and the functions of the individual components, as well as their connection to other components. It already includes the interface functionalities for accessing the CAD model, communication via XMPP, the interface to the agent management with the acquaintance and ontology database, and the integration of external resources and databases. In order to build a domain-specific assistance system from the template, this has to be equipped with specific capabilities, such as the determination of manufacturing-specific features from the CAD model, the storage of domain knowledge, such as relevant standards or design guidelines, and the definition of the reasoning behavior based on beliefs, plans and intentions. The MOKA approach can provide support here, as the *package* phase assists in the implementation of the formalized knowledge and the *activate* phase inserts the agent into the MAS and prepares it for use.

5. Application Example of the Agent Template

After presenting the architecture in section 4.2 and the implementation as a template in SPADE in section 4.3, the next step is to describe the application of the template using an example from engineering design. The CAD model of a mounting bracket, which is frequently used as a fixture in production engineering and quality assurance, is used as an application example. The raw part of the mounting bracket is cast from gray cast iron and then machined on a milling machine to ensure the required tolerances. To support the designer in decision-making, the part has to be evaluated for its manufacturability and adjusted accordingly. For this purpose, an interface to a CAD program, a knowledge base for manufacturing machines and tools as well as their manufacturing boundary conditions, and standardized communication between the agents are used [32].

To illustrate the application of the template, a simplified MAS consisting of a *manufacturing agent* and a *design agent* is presented for the evaluation of the example. For the definition of the agents, the Prometheus notation of Padgham and Winikoff [28] in figure 2 was used, which contains the percepts, actions, capabilities, plans, databases, and messages. The messages are further subdivided into the messages between the agents (dark green) and the messages between the agents and the designer or user (gray).

The objective of the *Manufacturing Agent* is to ensure the manufacturability of the CAD model. To do this, it reads out the faces and edges of the BRep model from the CAD model in Inventor and uses them to determine manufacturing features such as pockets, slots, and steps. In the next step, it checks the model for undercuts by projecting the base faces of pockets, slots, and steps onto the bounding box of the part, and then reviews these projections for overlaps. If an undercut is found, the agent first attempts to execute a plan that results in the elimination of the feature that leads to the undercut. Since this action may have consequences for the function of the part, the manufacturing agent initiates a request to the design agent. In the case of the application example, the design agent rejects the change request because the raw part is a casting and material accumulation should be avoided at this location. This rejection

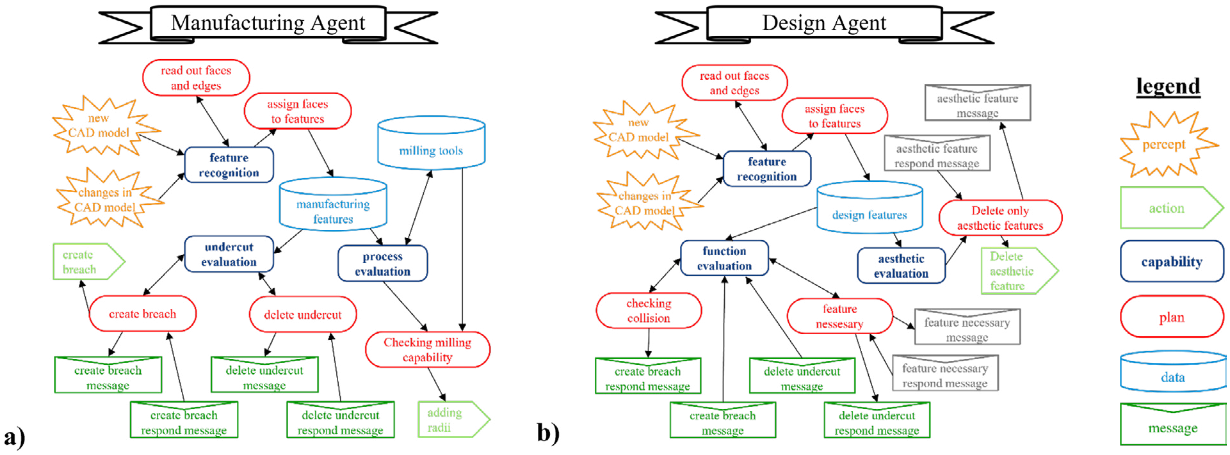

Fig. 2. Prometheus Agent Overview Diagram: (a) Manufacturing Agent; (b) Design Agent.

causes an error in the execution of the first plan, so the manufacturing agent has to pursue an alternative plan, which involves manufacturing the pocket from the outside of the part. After consulting with the design agent again and getting approval for the change, the manufacturing agent adjusts the feature in question in the CAD model. After the part has been checked for undercuts, the next step is to review the manufacturability of the pockets and slots using the stored milling tools. For this purpose, the side faces of the pockets and slots are checked for sharp edges and, if necessary, rounded depending on the diameter of the milling tool. This inference cycle is performed for all determined features in the part.

The *Design Agent* aims to ensure the functionality of the part, seeking consultation with the designer or user if contextual information is missing. The Design Agent also reads out the faces and edges of the CAD model and checks conical and convex faces for possible features such as fillets and chamfers. If the agent finds external fillets and chamfers, it uses an XMPP message to ask the designer whether they fulfill a function or were inserted for purely aesthetic reasons. As a result, purely aesthetic features are suppressed in the CAD model and not taken into account for further processing. When the manufacturing agent requests the deletion of features, the design agent asks the designer whether they are necessary. Furthermore, the Design Agent checks if a change of a feature affects other features. If this is not the case, the agent automatically gives permission for the change.

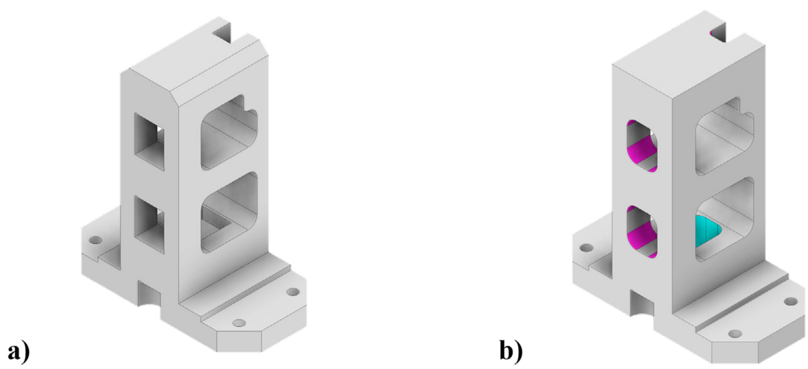

Fig. 3. Mounting Bracket as Application Example: (a) Unprocessed Part; (b) Revised Part.

The performed evaluation of the mounting bracket can be seen in figure 3. Here, the undercut was eliminated by extruding to the bottom of the part (cyan marked areas). In addition, for manufacturing reasons, the edges of the lateral pockets and the upper slot were rounded depending on the milling tools used (magenta marked areas). The chamfers on the top of the part were also removed, as they were only there for aesthetic reasons and would have cost unnecessary production time.

6. Discussion and Conclusion

To overcome the problem of lack of knowledge sharing, communication, and the resulting lack of holistic effort estimation or evaluation of design solutions between the manufacturing and design department, this paper presents the development of a knowledge-based and collaborative engineering design agent. This agent is capable of both an analytical evaluation of the design in terms of manufacturability and a synthetic application by automatically adapting the CAD model of the part. The combination of knowledge-based and multi-agent systems for analytical and synthetic application distinguishes the presented approach from related literature where only suggested changes are formulated or an adaptation of the CAD model is performed in closed solution spaces.

By using the CommonKADS method, the key properties of the system and its environment can be captured by building separate models so that specific properties such as agent communication and the formalization of expert knowledge are represented. Furthermore, the extension with the MOKA method can support an adaptation of the engineering design agent to specific domain tasks and the subsequent embedding into the multi-agent system.

The requirements derived from the specification could be fulfilled because the system is directly integrated into the design process through the link to a CAD program. In addition, the template can be used to represent various experts who have the ability to conduct negotiations and respond to the beliefs and intentions of other agents. The designer is directly involved in the decision-making process by being consulted for queries and the agent's inferences are documented in a comprehensible way. Furthermore, the use of the SPADE framework proves to be practical, as the Python programming language allows the use of a variety of AI libraries, and the communication via XMPP offers the possibility to interact with other agents, humans, and even third-party tools.

The presented development is part of ongoing research, where the prototype has been tested with CAD designs of students in mechanical engineering, who were asked to design milled components as an exercise, which were subsequently checked by the MAS and human experts to ensure the functionality of the MAS. So far single parts have served as an application example, which have simple mathematical geometries and whose machining is possible with a 2.5D milling machine. This results in the need for further research, as an assessment of the scalability and robustness of the system when the agents make autonomous decisions remain to be made. Furthermore, the use of the engineering design agent for the execution of virtual design reviews is desired, whereby the use of learning algorithms, such as case-based or probabilistic reasoning, will be examined.

References

[1] Bellifemine, F., Caire, G., Greenwood, D., 2007. Developing Multi-Agent Systems with JADE. Wiley Series in Agent Technology. 1 ed., John Wiley & Sons, Ltd, Chichester, UK. doi:10.1002/9780470058411.

[2] Bergenti, F., Caire, G., Monica, S., Poggi, A., 2020. The first twenty years of agent-based software development with JADE. Autonomous Agents and Multi-Agent Systems 34, 36. doi:10.1007/s10458-020-09460-z.

[3] Cass, S., 2020. The top programming languages: Our latest rankings put Python on top-again - [Careers]. IEEE Spectrum 57, 22–22. doi:10.1109/MSPEC.2020.9150550.

[4] De Hoog, R., Martil, R., Wielinga, B., 1994. The Common KADS model set. Technical Report February 1998. ESPRIT Project P5248 KADS-II.

[5] Doellken, M., Zimmerer, C., Matthiesen, S., 2020. CHALLENGES FACED BY DESIGN ENGINEERS WHEN CONSIDERING MANUFACTURING IN DESIGN – AN INTERVIEW STUDY. Proceedings of the Design Society: DESIGN Conference 1, 837–846. doi:10.1017/dsd.2020.302.

[6] Dostatni, E., Diakun, J., Grajewski, D., Wichniarek, R., Karwasz, A., 2016. Multi-agent system to support decision-making process in design for recycling. Soft Computing 20, 4347–4361. doi:10.1007/s00500-016-2302-z.

[7] Feldhusen, J., Grote, K.H. (Eds.), 2013. Pahl/Beitz Konstruktionslehre - Methoden und Anwendung erfolgreicher Produktentwicklung. Springer Berlin Heidelberg. doi:10.1007/978-3-642-29569-0.

[8] Feng, S.C., 2005. Preliminary design and manufacturing planning integration using web-based intelligent agents. Journal of Intelligent Manufacturing 16, 423–437. doi:10.1007/s10845-005-1655-4.

[9] Gembarski, P.C., 2020. On the Conception of a Multi-agent Analysis and Optimization Tool for Mechanical Engineering Parts, in: Jezic, G., Chen-Burger, J., Kusek, M., Sperka, R., Howlett, R.J., Jain, L.C. (Eds.), Agents and Multi-Agent Systems: Technologies and Applications 2020. John Wiley & Sons, Singapore. volume 186, pp. 93–102. doi:10.1007/978-981-15-5764-4{_}9.

[10] Gembarski, P.C., Plappert, S., Lachmayer, R., 2021. Making design decisions under uncertainties: probabilistic reasoning and robust product design. Journal of Intelligent Information Systems 57, 563–581. doi:10.1007/s10844-021-00665-6.

[11] Georgeff, M.P., Rao, A., 1992. An abstract architecture for rational agents, in: Proc. of the Third International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. pp. 439–449.

[12] Håkansson, A., Hartung, R., Moradian, E., Wu, D., 2010. Comparing Ontologies Using Multi-agent System and Knowledge Base, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). volume 6279, pp. 124–134. doi:10.1007/978-3-642-15384-6_14.

[13] Hopgood, A.A., 2021. Intelligent Systems for Engineers and Scientists. CRC Press, Boca Raton. doi:10.1201/9781003226277.

[14] Hvam, L., Mortensen, N.H., Riis, J., 2008. Product Customization. Springer Berlin Heidelberg. doi:10.1007/978-3-540-71449-1.

[15] Iglesias, C.A., Garijo, M., González, J.C., Velasco, J.R., 1996. A methodological proposal for multiagent systems development extending CommonKADS, in: Proceedings of the 10th Banff knowledge acquisition for knowledge-based systems workshop, pp. 21–25.

[16] Jennings, N.R., Wooldridge, M., 1995. Applying agent technology. Applied Artificial Intelligence 9, 357–369. doi:10.1080/08839519508945480.

[17] Jia, H.Z., Ong, S.K., Fuh, J.Y., Zhang, Y.F., Nee, A.Y., 2004. An adaptive and upgradable agent-based system for coordinated product development and manufacture. Robotics and Computer-Integrated Manufacturing 20, 79–90. doi:10.1016/j.rcim.2003.08.001.

[18] Jin, Y., Lu, S.C., 2004. Agent based negotiation for collaborative design decision making. CIRP Annals - Manufacturing Technology 53, 121–124. doi:10.1016/S0007-8506(07)60659-3.

[19] Kingston, J., 2001. Modelling Agents and Communication using CommonKADS, in: Research and Development in Intelligent Systems XVII. Springer London, pp. 301–319. doi:10.1007/978-1-4471-0269-4_22.

[20] Kratzer, M., Rauscher, M., Binz, H., Goehner, P., 2011. An agent-based system for supporting design engineers in the embodiment design phase. ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design 10, 178–189.

[21] Liu, Q., Cui, X., Hu, X., 2008. An agent-based intelligent CAD platform for collaborative design. Communications in Computer and Information Science 15, 501–508. doi:10.1007/978-3-540-85930-7_64.

[22] Madhusudan, T., 2005. An agent-based approach for coordinating product design workflows. Computers in Industry 56, 235–259. doi:10.1016/j.compind.2004.12.003.

[23] Mahesh, M., Ong, S.K., Nee, A.Y., 2007. A web-based multi-agent system for distributed digital manufacturing. International Journal of Computer Integrated Manufacturing 20, 11–27. doi:10.1080/09511920600710927.

[24] Medani, O., Ratchev, S., 2006. A STEP AP224 agent-based early manufacturability assessment environment using XML. The International Journal of Advanced Manufacturing Technology 27, 854–864. doi:10.1007/s00170-004-2279-0.

[25] Milton, N., 2008. Knowledge Technologies. Polimetrica, International Scientific Publisher.

[26] Nwana, H.S., Ndumu, D.T., Lee, L.C., Collis, J.C., 1999. Zeus: A toolkit for building distributed multiagent systems. Applied Artificial Intelligence 13, 129–185. doi:10.1080/088395199117513.

[27] Ortiz-Arroyo, D., Larsen, H.L., 2004. A Multi-Agent System Framework for Collaborative Decision Making Under Uncertainty, in: Symposium on Challenges in the Internet and Interdisciplinary Research, International IPSI-2004, IPSI.

[28] Padgham, L., Winikoff, M., 2004. Developing Intelligent Agent Systems. Wiley. doi:10.1002/0470861223.

[29] Palanca, J., Terrasa, A.A., Julian, V., Carrascosa, C., 2020. Spade 3: Supporting the new generation of multi-agent systems. IEEE Access 8, 182537–182549. doi:10.1109/ACCESS.2020.3027357.

[30] Plappert, S., Gembarski, P.C., Lachmayer, R., 2020. The Use of Knowledge-Based Engineering Systems and Artificial Intelligence in Product Development: A Snapshot, in: Światek, J., Borzemski, L., Wilimowska, Z. (Eds.), Advances in Intelligent Systems and Computing. Springer International Publishing, Cham. volume 1051, pp. 62–73. doi:10.1007/978-3-030-30604-5_6.

[31] Plappert, S., Gembarski, P.C., Lachmayer, R., 2021. Multi-Agent Systems in Mechanical Engineering: A Review, in: Jezic, G., Chen-Burger, J., Kusek, M., Sperka, R., Howlett, R.J., Jain, L.C. (Eds.), Agents and Multi-Agent Systems: Technologies and Applications 2021. Springer Singapore, Singapore. volume 241, pp. 193–203. doi:10.1007/978-981-16-2994-5{_}16.

[32] Plappert, S., Gembarski, P.C., Lachmayer, R., 2022. Knowledge-Based Design Evaluation of Rotational CAD-Models with a Multi-Agent System, in: Borzemski, L., Selvaraj, H., Świkatek, J. (Eds.), Advances in Systems Engineering, Springer International Publishing, Cham. pp. 47–56. doi:10.1007/978-3-030-92604-5{_}5.

[33] Roy, R., Hinduja, S., Teti, R., 2008. Recent advances in engineering design optimisation: Challenges and future trends. CIRP Annals - Manufacturing Technology 57, 697–715. doi:10.1016/j.cirp.2008.09.007.

[34] Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H., de Velde, W., 1994. CommonKADS: A comprehensive methodology for KBS development. IEEE Expert 9, 28–37.

[35] Skarka, W., 2007. Application of MOKA methodology in generative model creation using CATIA. Engineering Applications of Artificial Intelligence 20, 677–690. doi:10.1016/j.engappai.2006.11.019.

[36] Stokes, M., 2001. Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications. volume 3. Professional Engineering Publishing London.

[37] Sun, J., Zhang, Y.F., Nee, A.Y.C., 2001. A distributed multi-agent environment for product design and manufacturing planning. International Journal of Production Research 39, 625–645. doi:10.1080/00207540010004340.

[38] Ugurlu, S., Gerhard, D., 2017. Supporting development teams in the early stages of product development through DFX-based knowledge management system and communication platform. Proceedings of the International Conference on Engineering Design, ICED 6, 121–130.

[39] Verein Deutscher Ingenieure, 2004. VDI 2223:2004-01, Systematic embodiment design of technical products.

[40] Weyns, D., Michel, F., 2015. Agent Environments for Multi-agent Systems – A Research Roadmap, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). volume 9068, pp. 3–21. doi:10.1007/978-3-319-23850-0_1.