INTERNATIONAL DESIGN CONFERENCE – DESIGN 2020
https://doi.org/10.1017/dsd.2020.300

APPLICATION OF KNOWLEDGE-BASED ENGINEERING FOR TEACHING DESIGN KNOWLEDGE TO DESIGN STUDENTS

S. Plappert ✉, L. Hoppe, P. C. Gembarski and R. Lachmayer

Leibniz Universität Hannover, Germany

✉ plappert@ipeg.uni-hannover.de

Abstract

For an optimal preparation of mechanical engineering students for their future work life, the use of problem-based methods in design teaching is investigated. Therefore an intelligent tutoring system for computer aided design education will be developed, which can automatically evaluate computer aided design models of design students. A knowledge-based engineering system will be used to assistance the design students in the execution of design tasks. Using a practice-oriented example, the application and the advantages for teaching will be verified and discussed.

Keywords: intelligent tutoring system (ITS), knowledge-based engineering (KBE), design education, computer-aided design (CAD), problem-based learning (PBL)

1. Introduction

The education of engineers at universities does not sufficiently prepare graduates for their daily work (Yadav et al., 2010). The main problem, according to Mills and Treagust (2003), is that engineers are not able to combine their theoretical knowledge with practical requirements and implications after graduation. However, the development of new products and the application of existing knowledge to new situations is one of the most important activities of engineers (Perrenet et al., 2000). To prepare mechanical engineering students in a targeted way for the job market, it can be beneficial to support the students in the design education with additional learning material.

1.1. Motivation

To consolidate the knowledge acquired during education and to establish a connection to design in practice, the use of tasks with real design problems is examined. In this problem-based learning, students do not only acquire practical knowledge, they can also understand the relevance of what they have learned (Prince and Felder, 2006). One approach to university teaching is the implementation of design projects in which students are given real experience in engineering design (Dutson et al., 1997). The teachers assign a design project task, which is prepared and submitted by the students. Subsequently, an evaluation of the design performance of the students takes place. During the elaboration process outside the lecture, intermediate results cannot be checked and no hints for finding solutions can be given. The problem is that individual assistance cannot be provided in large lectures with several hundred students using traditional teaching tools (Negnevitsky, 1998). So, for the support of students in design education, an information system is desirable which can check the learning progress of students during the design process and automatically evaluate the design solutions of a large number of engineering students. The

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

advantages of an information system are that many students can intensify what they have learned from the lectures at their own speed of learning and work on the tasks free of spatial and time constraints (Hwang, 2003). Teaching can also benefit from this, as an evaluation of the processed tasks can be carried out by the information system and thus a direct feedback for the lecture can be generated.

This paper discusses how a knowledge-based engineering system (KBES) can be used to support students in their design tasks. For this purpose, the architecture of an intelligent tutoring system (ITS) for design education was developed and verified using an example from teaching.

1.2. Structure of the paper

In the following section 2, the paper describes the technical basics of intelligent tutoring systems and knowledge-based engineering systems. In section 3 the requirements for an intelligent tutoring system for design education are discussed. The resulting architecture of an intelligent tutoring system with its components is described in more detail in section 4. Therefore, it is explained and tested with an example from design education. The concluding section 5 summarizes the paper and gives an outlook for future work.

2. Related work

2.1. Intelligent tutoring systems (ITS)

In order to simulate a human teacher on an e-learning platform, an intelligent tutoring system (ITS) can be used for individual instructional delivery. An ITS is an educational software that actively supports the student in learning by using artificial intelligence (AI) to track the student's particular performance in order to provide tailored feedback (Chakraborty et al., 2010). In engineering sciences, ITS are already occasionally used for teaching purposes. For example, Negnevitsky (1998) uses ITS in electrical engineering to teach fault analysis. Therefore an object oriented tool is used which guides the student through the exercises with a graphical user interface (GUI). The student can choose from three sets of exercises, which vary in difficulty. The editing of the exercises is realized by entering a numerical value that is calculated by the student. When an incorrect solution is entered, the two-step tutoring system applies. At the first incorrect entry, the student is only informed that his solution is incorrect and that he has to check his calculation. If the input is incorrect for a second time, the calculation of the correct solution is revealed to the student. This type of tutoring system has only one correct solution.

In the area of manufacturing engineering, an ITS is used to develop a tutorial on how to use a computer numerical control (CNC) machine (Hsieh and Hsieh, 2001). The ITS is also structured as a step-by-step tutorial and has the goal to prepare the students for the programming of the machine before using the CNC machine in the practical course. Thus, the theoretical basis can be already established before the practical course and the students can gain more experience with the used equipment of the machine. During the tutorial, screenshots of the GUI of the machine, important supplementary information on control and test questions is used. The knowledge-based tool is used to generate the presentation of the tutorial, so that changes can be easily made in the knowledge base without having to adapt the user interface as in traditional tutorials.

For die design, Cheok and Nee (1997) use an ITS with a computer-aided design and drafting (CADD) system as the front-end. By the possibility of the configuration of the die design many different design solutions can be generated by the change of input parameters. This allows the student to represent and evaluate different configurations. In order to give the student an orientation towards the desired solution, an explanation window is displayed additionally to the shown configuration. In addition with the help of the explanation and guidance capabilities, the program explains the stored decisions when selecting a solution. The expertise stored in a knowledge-based system (KBS) can also be used to provide knowledge during education, so that a similarity between ITS and KBS can be identified (Cheok and Nee, 1997). The stored knowledge is used to make inferences from new findings and provide the user the result.

2.2. Knowledge-based engineering systems (KBES)

Knowledge-based systems (KBS) apply expert knowledge in a computer program with the use of artificial intelligence (AI) techniques to solve complex problems (Milton, 2008). According to

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

Hopgood (2016), the essential components of a KBS system include the knowledge base, the inference engine and the interface to the environment (Figure 1):

1. Knowledge base: Explicit knowledge is stored in the knowledge base in the form of information and rules.
2. Inference engine: The inference engine controls the use of knowledge from the knowledge base.
3. Interface to the environment: The interface establishes the interaction between the KBS and the environment.

According to Milton (2008), a knowledge-based engineering system (KBES) is a special kind of KBS, which is extended by computer aided analysis (CAE) and computer aided design (CAD) capabilities. KBES are particularly suitable for mapping design solution spaces, because expert knowledge can be documented with them and an interface to CAD is already provided. Usually the designer is supported by KBE or AI during the embodiment and detail design phase (Plappert et al., 2020). The stored knowledge can either be programmed with rules or represented in table form. Among other things, standard part catalogues and design rules can be easily implemented. These properties make it possible to systematically represent even complex tasks. A further advantage of KBES in education is the simple extensibility of the knowledge base, so that new teaching content can be added quickly. In addition, engineering decisions can be identified and procedures for model design clarified (Verhagen et al., 2012).

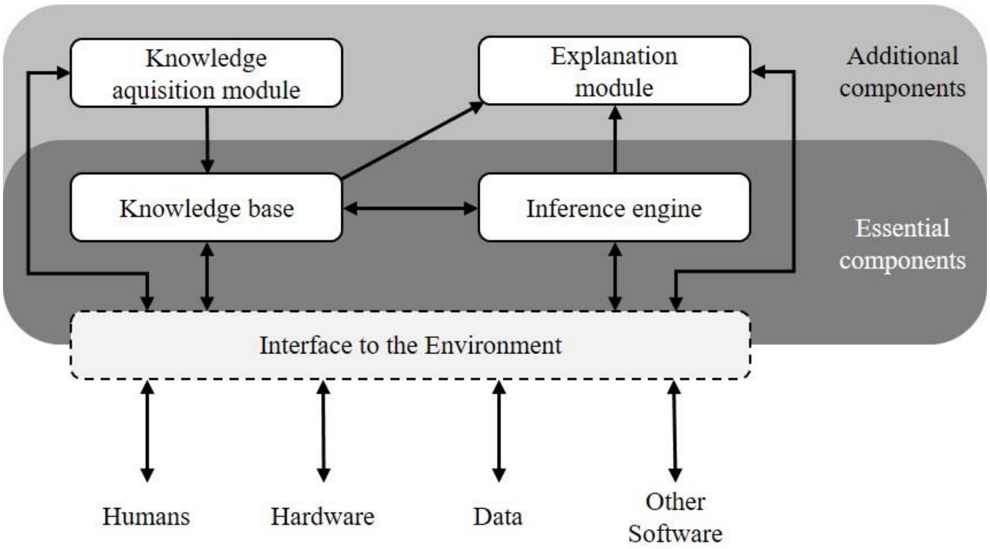


Figure 1. Components of a knowledge-based system (Hopgood, 2016)

As additional components for a KBS a knowledge acquisition module and an explanation module can be used. The explanation module explains the decisions of the KBS so that the user is able to review them. Usually, the explanation module assists the user in entering control parameters by hints to the restrictions or provides a justification for the solution of the KBS.

All ITS considered in section 2.1 use tasks for which the solution can be fully stored. Therefore users are often guided to a specific solution with an ITS, which is structured as a tutorial with feedback. An exception to this is the ITS of Cheok and Nee (1997), where the user can select configurations in a CADD system and receives feedback on the validity of the selected solution. The disadvantage of this ITS is that the user can only try out ready-made configurations, but cannot develop a design solution himself and model it in a CAD system.

3. Requirements for an ITS for design education

The curriculum for mechanical engineering students provides for the early semesters a basic understanding of product development. For this purpose, students at the Leibniz University Hannover learn about different design elements and how machine elements and standard parts are used in a functional and production-oriented manner. Therefore, the students should understand how the torques

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

of gears are converted by gear pairs and transmitted to the shafts, e.g. by using keyways. In addition, they should be able to identify how different types of bearings absorb the generated forces and enable the shaft to rotate. For this purpose, the students should recognize where the functional surfaces of the shaft are located and which design elements are required to enable the mounting and retaining of bearings and gear wheels. The students are taught practical skills in design projects, parallel to the theoretical content of the lecture. In these design projects, for example, a shaft is drawn by hand or parts and assemblies are modelled in the CAD program Autodesk Inventor Professional.

In order to ensure that the learning objectives required by the curriculum can actually be achieved by an ITS, the method of constructive alignment is applied. This concept was developed to plan lectures and to coordinate them with regard to their *learning outcomes*, *teaching and learning activities* and *assessment*. The linking of these points ensures that the learning outcomes correspond to the activities of teaching and that the assessment examines exactly the desired learning outcomes (Biggs, 2003).

The *learning outcomes* describe the competences that students have acquired after the lecture. After the first two lectures of design education and the related design projects, students should be able to independently create part and assembly models using design rules and standards in a CAD system. For this purpose, a confident handling of a CAD system, knowledge of design methods and the proper use of machine elements are required.

The *teaching and learning activities* describe the approach to achieve the set learning outcomes in the best possible way. Problem-based learning is particularly suitable here, in which students are confronted with open, poorly structured and real-world problems (Prince and Felder, 2006; Zumbach, 2003). In order to implement this type of learning method, ITS can provide students with design tasks, independent of time and place. When working on these tasks, feedback must be given, which contributes to deepening design knowledge by providing hints.

The *assessment* represents a situation in which the desired learning outcomes can be verified. An answer-choice technique, commonly used for tutoring systems, is unsuitable for design tasks because, on the one hand, it does not motivate students to learn sustainably (Biggs and Tang, 2007). On the other hand, there is usually not only one solution for design tasks, but several solutions that meet the requirements. In this way, the task definition can be regarded as a requirement space, which contains the technical development goals and product properties. Opposite this is the solution space, which includes all possible solutions for implementing the requirements in a product (Ponn and Lindemann, 2011; Gembarski and Lachmayer, 2017). In order for the ITS for design education to be able to individually and objectively evaluate the students' design solutions and act independently of time and location, the correction has to be automated.

From the learning objectives of the curriculum and the constructive alignment, the following main requirements for an ITS in design education are identified:

- Setup of a solution space: Since there is not only one solution for problem- and practice-oriented tasks, the space of all possible solutions must be represented within the ITS
- Feature recognition: In order to be able to automatically evaluate the students' design solutions, the design elements of the CAD models have to be analysed
- Explanation module: In order to provide direct feedback to students, possible errors must be categorized and evaluated for their similarity to the valid solution. In addition, hints that indicate the correct solution have to be assigned to the errors.
- Knowledge base: To be able to check the use of standards and standard parts, these must be stored in the form of tables or design rules.

4. Application example in teaching

In this section, the architecture of the ITS for design education is described using the example of a shaft design (Figure 2). Therefore a task definition was developed, which is oriented to the curriculum of the design education. Based on the task definition, the interface between the CAD system and the KBES is explained. The development of the KBES including the feedback function will then be discussed. Additionally, the process of applying the ITS based on the task definition is summarized and the verification of the ITS for design education is explained using a test case.

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

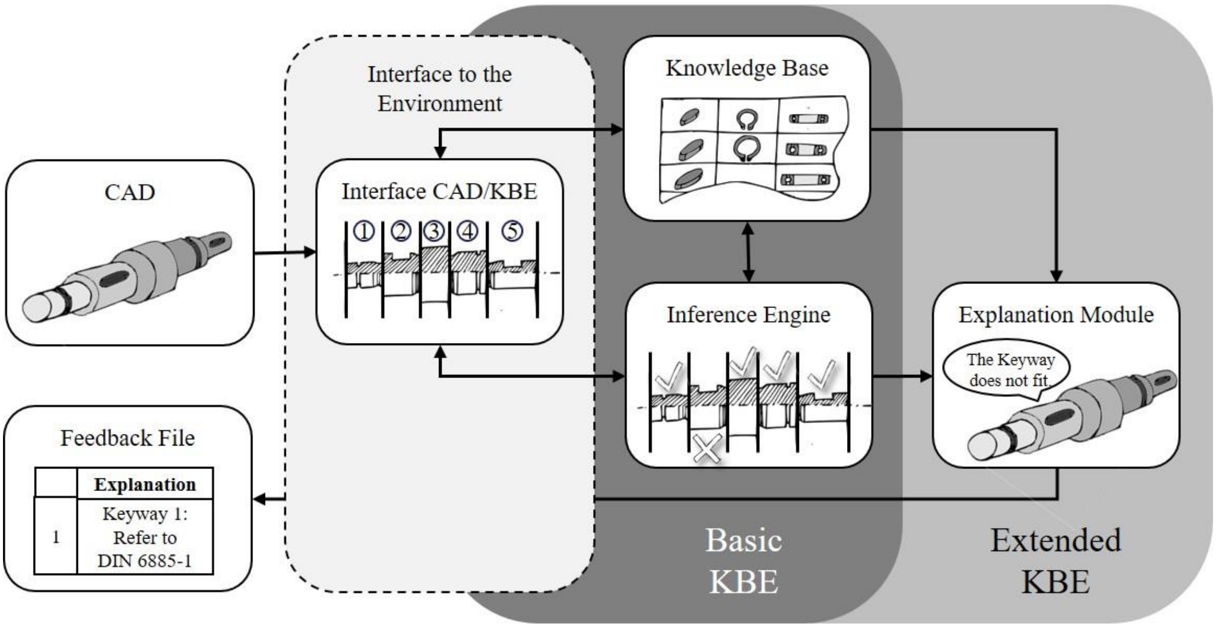


Figure 2. Architecture of an Intelligent Tutoring System for Design Education

4.1. Task definition of the application example

In the first application, the students are asked to design a gear shaft, which is a good example to apply what they have learned from the lectures (Figure 3). The students have to use machine elements and have to create the design on the basis of standards. To reduce the programming effort, the task definition was limited, which also results in a limitation of the possible number of solutions.

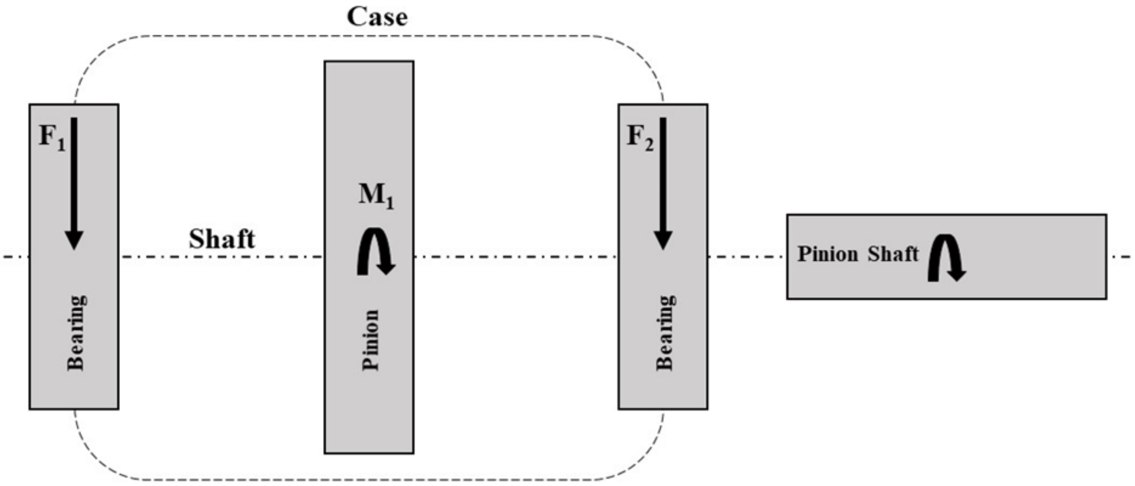


Figure 3. Task from the apprenticeship for a gearshaft

The task describes a gear shaft which is connected to a gear housing by two bearings. In the housing, the shaft is rotated by a gear wheel and transmits the torque to the pinion shaft outside the housing. The students should design a shaft suitable for production, on which two bearings and two gears can be assembled. In order to limit the number of possible solutions and to keep the creation effort of the KBES controllable, the task is provided with additional restrictions. For example, the variants of standard parts are given to the student and the loads of the bearings are assumed only as radial forces. Based on the task, the students have to create a CAD model of the shaft. To secure the bearings, grooves for the circlips on the bearing seats of the shaft are necessary. The torques should be transmitted with keyways, which are dependent on the shaft diameter. In addition, undercuts have to be planned for easy assembly of the bearings and gears.

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

4.2. Evaluation of the design solution

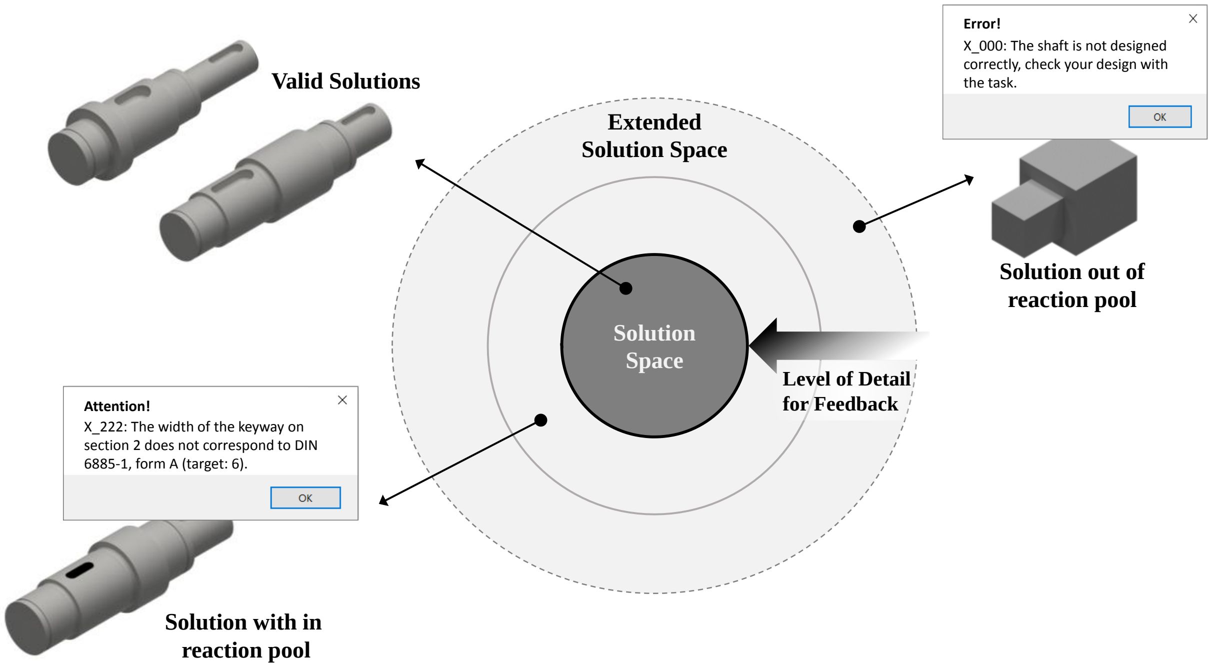
The ITS for design education requires an interface between the CAD program and the KBES. In order to verify the student's design generated by the KBES, the results obtained have to be interpreted. For this, the CAD model of the student is analysed and the relevant parameters are transferred to the KBES. The CAD program Autodesk Inventor Professional was used to model the application example, as the students are already familiar with it from the lectures and have used it for the design project. The procedure described in this paper can also be implemented in other CAD programs, e.g. Skarka (2007) shows how a KBES can be set up within the CAD program CATIA.

For the analysis of the CAD model, a visual basic for applications (VBA) macro was used to access the application programming interface (API) of Autodesk Inventor Professional. The VBA macro recognizes the relevant elements of the shaft similar to a feature recognition program. Thus, the surfaces of the CAD model can be read and interpreted with the VBA macro. The shaft heels, grooves, keyways and undercuts are recognized as features. In addition, the basic dimensions of the shaft and the geometry of the features are detected and stored as parameters. All parameters are then transferred to the KBES via the Inventor parameter table.

The KBES reads the transferred parameters of the interface CAD/KBE from the parameter table and enters them into a GUI. In the iLogic code the restrictions from the task as well as the access to the standard part catalogues are programmed. iLogic is a programming language that can be used to document design rules and link parameters (Gembarski, 2019). The standard part catalogues of keyways, circlips and deep groove ball bearings are embedded in Inventor as Excel tables. This has the advantage that the knowledge base can be easily expanded and updated. The stored knowledge from the restrictions and the standard part catalogues is also used to program iLogic rules for user-specific feedback.

4.3. Generation of feedback with an explanation module

To keep the programming effort within limits, while at the same time supporting the students, by providing feedback, towards a valid solution, the ITS distinguishes between three types of possible solutions, depending on their distance from the solution space (Figure 4).



Error!
X_000: The shaft is not designed correctly, check your design with the task.

Attention!
X_222: The width of the keyway on section 2 does not correspond to DIN 6885-1, form A (target: 6).

Figure 4. Presentation of the extended solution space with the application example

Different feedback is given for each type of possible solution. *Solution out of reaction pool*, for example a cube-shaped design, does not correspond to the basic geometry of a valid shaft, so the feedback function will give an error code. Close *solution with in reaction pool* corresponds to the

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

basic geometry of a valid shaft. However, either required elements are not present on the shaft section, neither the dimensions of the shaft section and elements are not included in the solution space. In this case, the user is shown the errors in the design and assistance is given which refer to standards and technical books. Thus the user is guided step by step to the design of a valid shaft. *Valid solution* fully corresponds to a solution of the solution space, so the user gets a positive feedback for the error-free solution. The feedback should not only help to reach the solution space. Rather, it should help the students to internalize the use of machine elements and standard parts during the solution process. This procedure also supports the user in finding his or her individual solution and motivates him or her to acquire personal expert knowledge.

4.4. Application of the ITS for design education

The restrictions of the task definition are already entered in the KBES and the explanation module can provide the appropriate feedback. Thus the task can be distributed by the teacher and the students can work on it directly. The process flow for the execution of the task by the student, the evaluation of the design solution by the ITS and the information transfer back to the teacher is shown in Figure 5.

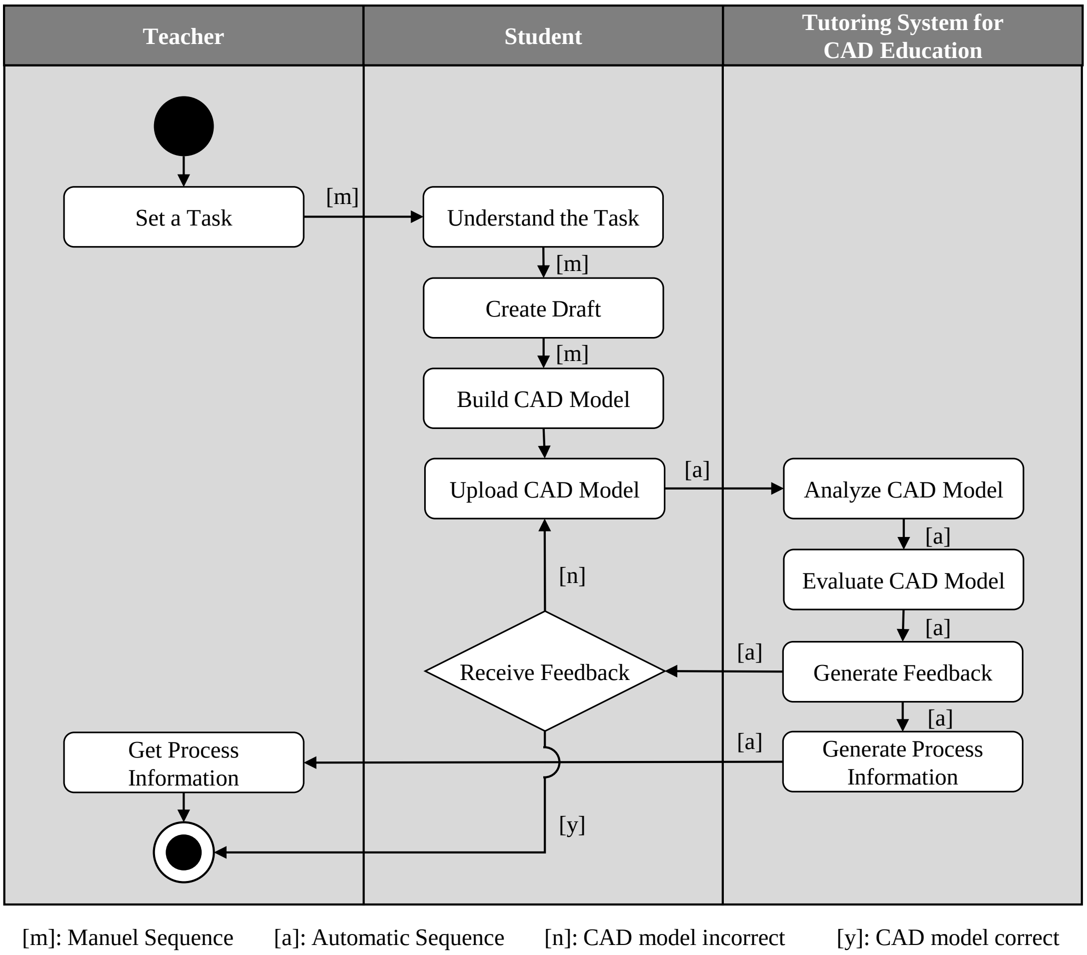


Figure 5. Process flow for the application of the ITS for design education

In the first step, the students should take their time to understand the task and to plan the next steps. The students then create a draft. In the next step, the CAD model is created, making sure that standard parts are used. Once the students have completed their design solution, they upload the CAD model to a university server so that the CAD model can be analysed by the ITS. After the ITS has evaluated the design solution, feedback is generated for the student and stored on the university server. In addition, the teacher receives feedback on the extent to which the tasks have been completed and which gaps in the students' knowledge still need to be closed.

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

4.5. Verification of the ITS for design education

For verification of the described system architecture of the ITS, the shaft design was verified with several test cases. For this purpose different shafts with different errors were created and modelled in the CAD program Autodesk Inventor Professional. To evaluate the design, the CAD data was manually transferred to the ITS. The VBA macro recognized the existing shaft geometry including the standard parts used and read out all the necessary dimensions of the shaft geometry. The parameters were then automatically transferred to the KBES. Subsequently, all errors in the design, such as the non-standard design of the undercuts, were saved in a text file. An exemplary test case for the verification is shown in Figure 6, which also contains error codes for feedback.

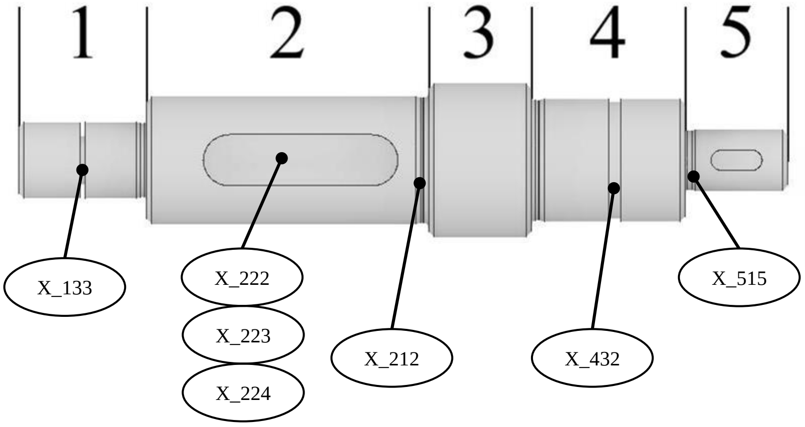


Figure 6. Test case for the verification of the ITS for design education

When evaluating the student's shaft, it is divided into sections, which on the one hand makes it easier to assign the functional surfaces to the task. On the other hand, this also helps to improve the traceability for feedback. The feedback consists of two parts. The first part of the feedback describes the error and the location of the error, through references to the section and the affected design element. The second part of the feedback contains hints for the student, such as the standard to be used or the target value in case of an erroneous parameter. The error codes with feedback for the test case are listed in Table 1.

Table 1. Error codes and feedback for the test case

codes	feedback
X_133	The diameter of the groove for circlip on section 1 does not correspond to DIN 471 (target: 14.3).
X_212	The radius of the undercut on section 2 does not correspond to DIN 509, form E, row 1 (target: 0.8).
X_222	The width of the keyway on section 2 does not correspond to DIN 6885-1, form A (target: 6).
X_223	The depth of the keyway on section 2 does not correspond to DIN 6885-1, form A (target: 3.5).
X_224	The length of the keyway on section 2 does not correspond to DIN 6885-1, form A.
X_432	The width of the groove for circlip on section 4 does not correspond to DIN 471 (target: 1.3).
X_515	An additional radius is required for the undercut on section 5 (see DIN 509).

The verification shows that the ITS elements are capable of automatically analysing the CAD model, identifying standard parts and functional surfaces and checking their correctness. In addition, the feedback indicates errors and points out their correction. In the future, the output of the task, the transfer of the CAD data and the providing of feedback are to be automated by an online platform.

5. Conclusion and future research

In this paper, an architecture for an ITS was developed, which supports the students in the design education as an additional tool. This system can be used for learning design methods and their application in a CAD system. In order to ensure that the learning objectives of the curriculum are

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

achieved, the requirements that the ITS has to meet were developed with the help of constructive alignment. The ITS allows the automatic evaluation of problem- and practice-oriented tasks for design education with different solution variants and supports the design process with feedback. The ITS was then verified according to the requirements using test cases. This showed that the described architecture can be used to create an ITS that can fulfil the requirements from design education.

In order to be able to conduct a targeted further development of the ITS for design education, validation under real conditions within a lecture is necessary. A subsequent evaluation by the students also increases acceptance. The ITS described in this paper includes the evaluation of CAD models of individual parts. In order to support the students in the design of assemblies, the extension of the ITS with regard to this option should also be examined. For large assemblies and complex designs, the extent to which AI methods can be used to evaluate CAD models in order to make the ITS for design education more efficient should be explored.

References

Biggs, J.B. (2003), *Teaching for quality learning at university*, McGraw-hill education, Buckingham, UK.

Biggs, J. and Tang, C. (2007), *Teaching for quality learning at university*, Vol. 3, Open University Press, New York.

Chakraborty, S., Roy, D. and Basu, A. (2010), “Development of knowledge based intelligent tutoring system”, *Advanced Knowledge Based Systems: Model, Applications & Research*, Vol. 1, pp. 74-100.

Cheok, B.T. and Nee, A.Y.C. (1997), “Developing a design system into an intelligent tutoring system”, *International Journal of Engineering Education*, Vol. 13 No. 5, pp. 341-346.

Dutson, A.J. et al. (1997), “A review of literature on teaching engineering design through project-oriented capstone courses”, *Journal of Engineering Education*, Vol. 86 No. 1, pp. 17-28. https://doi.org/10.1002/j.2168-9830.1997.tb00260.x

Gembarski, P.C. (2019), *Komplexitätsmanagement mittels wissensbasiertem CAD: Ein Ansatz zum unternehmenstypologischen Management konstruktiver Lösungsräume [PhD Thesis]*, Leibniz University of Hannover, TEWISS.

Gembarski, P.C. and Lachmayer, R. (2017), “Designing Customer Co-Creation: Business Models and Co-Design Activities”, *International Journal of Industrial Engineering and Management (IJIEM)*, Vol. 8 No. 3, pp. 121-130.

Hopgood, A.A. (2016), *Intelligent systems for engineers and scientists*, CRC press.

Hsieh, S.-J. and Hsieh, P.Y. (2001), “Intelligent tutoring system authoring tool for manufacturing engineering education”, *International Journal of Engineering Education*, Vol. 17 No. 6, pp. 569-579.

Hwang, G.-J. (2003), “A conceptual map model for developing intelligent tutoring systems”, *Computers & Education*, Vol. 40 No. 3, pp. 217-235. https://doi.org/10.1016/s0360-1315(02)00121-5

Mills, J.E. and Treagust, D.F. (2003), “Engineering education—Is problem-based or project-based learning the answer”, *Australasian journal of engineering education*, Vol. 3 No. 2, pp. 2-16.

Milton, N.R. (2008), *Knowledge technologies*, Polimetrica sas.

Negnevitsky, M. (1998), “A knowledge based tutoring system for teaching fault analysis”, *IEEE transactions on power systems*, Vol. 13 No. 1, pp. 40-45. https://doi.org/10.1109/59.651611

Perrenet, J.C., Bouhuijs, P.A.J. and Smits, J. (2000), “The suitability of problem-based learning for engineering education: theory and practice”, *Teaching in higher education*, Vol. 5 No. 3, pp. 345-358. https://doi.org/10.1080/713699144

Plappert, S., Gembarski, P.C. and Lachmayer, R. (2020), “The Use of Knowledge-Based Engineering Systems and Artificial Intelligence in Product Development: A Snapshot”, In: Świątek, J., Borzemski, L. and Wilimowska, Z. (Eds.), *Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology - ISAT 2019*, Springer International Publishing, Cham, pp. 62-73. https://doi.org/10.1007/978-3-030-30604-5_6

Ponn, J. and Lindemann, U. (2011), *Konzeptentwicklung und Gestaltung technischer Produkte: systematisch von Anforderungen zu Konzepten und Gestaltlösungen*, Springer-Verlag.

Prince, M.J. and Felder, R.M. (2006), “Inductive teaching and learning methods: Definitions, comparisons, and research bases”, *Journal of Engineering Education*, Vol. 95 No. 2, pp. 123-138. https://doi.org/10.1002/j.2168-9830.2006.tb00884.x

Skarka, W. (2007), “Application of MOKA methodology in generative model creation using CATIA”, *Engineering Applications of Artificial Intelligence*, Vol. 20 No. 5, pp. 677-690. https://doi.org/10.1016/j.engappai.2006.11.019

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press

Verhagen, W.J.C. et al. (2012), "A critical review of Knowledge-Based Engineering: An identification of research challenges", *Advanced engineering informatics*, Vol. 26 No. 1, pp. 5-15. https://doi.org/10.1016/j.aei.2011.06.004

Yadav, A., Shaver, G.M. and Meckl, P. (2010), "Lessons learned: Implementing the case teaching method in a mechanical engineering course", *Journal of Engineering Education*, Vol. 99 No. 1, pp. 55-69. https://doi.org/10.1002/j.2168-9830.2010.tb01042.x

Zumbach, J. (2003), *Problembasiertes Lernen*, Waxmann Verlag.

https://doi.org/10.1017/dsd.2020.300 Published online by Cambridge University Press