Four Independent Knowledge Domains to Enable an Agile, Distributed Development of User-Centred Engineering Configurators

Schäffer, Eike; Mayr, Andreas; Reichenstein, Tobias; Shafiee, Sara; Franke, Jörg

Published in:
Procedia CIRP

Link to article, DOI:
10.1016/j.procir.2021.10.021

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Four Independent Knowledge Domains to Enable an Agile, Distributed Development of User-Centred Engineering Configurators

Eike Schäffera,*, Andreas Mayra, Tobias Reichensteinb, Sara Shafieeb, Jörg Frankea

aInstitute for Factory Automation and Production Systems, Friedrich-Alexander University Erlangen-Nuremberg, Egerlandstr. 7, Erlangen 91058, Germany
bTechnical University of Denmark, Department of Mechanical Engineering, 2800 Kgs. Lyngby, Denmark

* Corresponding author. Tel.: +49 9131 85-28972; fax: +49 9131 302528. E-mail address: Eike.Schaeffer@faps.fau.de

Abstract

Robot-centric automation solutions (RAS) promise greater efficiency and consistent quality in production, relieving workers of physically demanding and dangerous tasks, especially in the times of COVID-19. Nevertheless, due to their relatively high complexity and implementation costs, RAS are only used to a limited extent by small and medium-sized manufacturing companies. As a rule, the high costs of RAS arise from custom engineering efforts, which take up to 70 percent of the acquisition costs. For this reason, it is necessary to optimise the engineering of RAS. However, software tools such as configurators have been used primarily for the individualisation of products, such as automobiles or clothing, based on variants predefined by the manufacturer, and less for the engineering of automation solutions. The development of knowledge-based systems, in particular knowledge-based engineering configurators (EC), is usually performed by few proficient experts with high development effort. One of the primary challenges in the knowledge acquisition is that several experts possess partial aspects of knowledge in an inhomogeneous, implicit form. Furthermore, there is a lack of efficient development methods for EC. By reusing knowledge elements from previous development projects, a sustainable increase in efficiency is possible. In order to enable an efficient development process of EC, we introduce a structuring model consisting of four knowledge domains (KD): knowledge about specific business cases (KD1), Best Practices as case-specific solution knowledge (KD2), logical expert knowledge (KD3) as well as semantically consistent data models for interoperability of different IT systems (KD4). As the four KD are independent, their development can be agilely divided among several teams or companies. Finally, the agile development approach is validated individually for each KD as well as comprehensively within the scope of the ROBOTOP platform for planning RAS.

© 2021 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the 9th CIRP Global Web Conference – Sustainable, resilient, and agile manufacturing and service operations: Lessons from COVID-19 (CIRPe 2021)

Keywords: Best Practice; configurator; knowledge-based; knowledge acquisition; KBEC; engineering; planning; division of labour; automation; robot

1. Introduction

The digitalisation of engineering promises automated workflows, higher speeds and reduced costs in the development of automation solutions. Due to large engineering efforts and high costs for system integrators, many robot-centric automation solutions (RAS) can currently not be economically realised. The impact of COVID-19 on global supply chains revealed the importance of local production facilities to ensure the supply of convenient goods, even in times of crisis. In high-wage countries such as Germany, however, economic production can only be sustained if the degree of automation is sufficiently high. To facilitate the development of automation solutions, better digital tools such as virtual reality (e.g. [1–3]), simulation (e.g. [4,5]) and configuration (e.g. [6–9]) as well as corresponding development methods are needed. In particular,
knowledge-based technologies such as configurators, being a subset of artificial intelligence (AI), promise high potential in this area [10–12].

Thus, the overall objective of this paper is to sharpen the understanding and possibilities of knowledge-based technologies within the engineering domain. Above all, knowledge-based engineering configurators (EC), also known as KBEC [13], promise a significant increase in efficiency in engineering [14]. However, their development requires a dedicated approach, which is addressed in this paper by introducing four independent knowledge domains, enabling an agile, distributed development of user-centred EC.

2. Relevant basics and related work

The following section first presents the basic conditions for factory planning. Building on this, knowledge-based approaches for addressing current challenges are presented.

2.1. Digital planning within factory and product life cycle

According to Westkämper [15], the factory and product life cycles (Fig. 1) are suitable for classifying the parallel activities of product development and factory planning. With each phase of the two life cycles, the degree of planning maturity and also the knowledge about customer requirements and possible solutions are increased up to a fully functional product including its production system. [13,15]

![Fig. 1. Product and factory life cycles connect within production phase [16]](image)

The use of digital plant, product or environment models, for example, create the prerequisite for effective assistance of the planner by context-related software applications [15]. These applications provide the bridge between the digital and real world. The underlying digital models are created and updated by real world data, if available, and can be used for simulative prediction of different scenarios. In this context, simultaneous engineering, also highlighted in Fig. 1, represents the integrated and time-parallel development of product and processes with the aim of shortening the duration from product idea to market launch ("time-to-market"), reducing development and manufacturing costs and improving product quality in a holistic sense [4].

2.2. Types of knowledge-based tasks

In general, knowledge-based tasks can be classified according to the type and degree of support (Fig. 2) [17]. In order to drive automation at manufacturing companies by means of RAS, there is a lack of suitable software tools to reduce the associated complexity and development effort. For this purpose, the concept of EC was introduced [13]. In the context of the general tasks of knowledge applications, EC can be classified as a specific activity within the subtask of design. In doing so, the EC occupies a hybrid position between the activities of planning and configuration. [17–19]

![Fig. 2. General tasks of knowledge applications condensed from [17–19] with supplementary classification of engineering configurators [14]](image)

2.3. Knowledge-based configurators as subset of artificial intelligence

Although there is no uniform definition, AI can generally be divided into several subareas: knowledge representation, automatic reasoning, machine learning, cognition, neuronal networks, image processing (also known as pattern recognition), language processing and expert systems (Fig. 3). Configurators represent a subset of expert systems and are often associated with construction. In general, the term "construction" refers to the individual, gradually changed stages within the design process, also called (product) configuration. The design process comprises the sum of all individual design steps. In the context of this work, configurations are understood as possible final solution alternatives. [10–12]

![Fig. 3. Classification of configuration within expert systems as a subarea of artificial intelligence [10,20,21]](image)

Classical configurators such as product configurators (PC) provide automatic access to various existing product variants [22]. With PC, the main value proposition of the product is usually clear and does not change significantly through configuration, unless performance or additional functionalities of a product are changed. Accordingly, most existing
configuration tools and methods have been developed for PC [23] building upon a modular product structure [24].

2.4. Product vs. knowledge-based engineering configurator

In RAS planning, however, the individual components are supplied by different manufacturers, usually without comprehensive standardisation (Fig. 4). This is where the concept of EC becomes relevant [13,14]. Compared to PC, EC are significantly more complex due to an indefinite number of manufacturers with different standards and product variants [24,25], which must be configured into a consistent automation solution [13].

The EC concept is divided into two general phases: First, the requirements are classified and a roughly matching automation solution that has already been implemented, a so-called Best Practice (BP), is selected as a starting point. This is followed by the individualisation of the BP by means of configuration functionalities, i.e. a customisation configuration. [26]

Thereby, experience-based knowledge can reduce costly and complex planning cycles. Furthermore, a user-centred approach plays a key role in reducing the operational and cognitive effort for users [27,28]. The application of user-centred development to configuration projects requires a complementary thinking style that extends the collaboration abilities of domain experts and configuration teams to make their outcome more innovative [29].

2.5. Need for action

Up to now, configurators have been used primarily for the individualisation of products, such as automobiles or clothing, on the basis of the variants predefined by the manufacturer, and less for the engineering of automation solutions. One of the primary challenges in the development of knowledge-based configurators lies in the knowledge acquisition, as usually several experts possess partial aspects of knowledge in an inhomogeneous, implicit form [10,30]. To enable an agile, user-centred development process of EC with scalable division of labour, a holistic concept is necessary.

3. Enabling an agile development of EC through the distinction of four independent knowledge domains

In the following, an approach is introduced for developing EC in an agile way. The individual knowledge elements are to be prepared in a modular form so that they can be reused in similar configuration projects. Therefore, the presented structuring model is divided into four independent knowledge domains (KD) for developing EC (Fig. 5):

- **KD$_1$** - Business cases (application need and benefit)
- **KD$_2$** - Best Practices (case-specific RAS solution knowledge)
- **KD$_3$** - Logical expert knowledge (generic RAS knowledge)
- **KD$_4$** - Semantically consistent data models (interoperability)

In this context, the term "independence" is not understood as the stochastic independence from probability theory, but as a sufficient separation of concerns allowing parallel development activities. In doing so, the individual KD can be developed in parallel, e.g. by various teams using Scrum [31] or another agile framework. Within Scrum, each KD can be broken down into individual items within the sprint planning and managed via Scrum "product backlog" of the aspired EC.

Fig. 4. Differentiation of product configurators and knowledge-based engineering configurators introduced by Schäffer et al. [13,14]

The EC concept is divided into two general phases: First, the requirements are classified and a roughly matching automation solution that has already been implemented, a so-called Best Practice (BP), is selected as a starting point. This is followed by the individualisation of the BP by means of configuration functionalities, i.e. a customisation configuration. [26]

Thereby, experience-based knowledge can reduce costly and complex planning cycles. Furthermore, a user-centred approach plays a key role in reducing the operational and cognitive effort for users [27,28]. The application of user-centred development to configuration projects requires a complementary thinking style that extends the collaboration abilities of domain experts and configuration teams to make their outcome more innovative [29].

Fig. 5. Structuring model with four generic knowledge domains (KD) for the development of user-centred EC as extension of [14,32]

- **KD$_1$** - Business cases (application need and benefit)
- **KD$_2$** - Best Practices (case-specific RAS solution knowledge)
- **KD$_3$** - Logical expert knowledge (generic RAS knowledge)
- **KD$_4$** - Semantically consistent data models (back-end IT architecture and preparation concepts for the EC via logical partial prototypes)

KD$_1$ focusses on the business value and user interaction perspective. The visual, digital modelling of realised, already implemented RAS with their structure, parts and conceptual material flow is addressed by KD$_2$. The necessary logical interdependencies for optimal combinations and solutions, which are modelled by means of constraints, are part of KD$_3$. The modeing done within KD$_4$ refers to the AI aspect of the EC as a kind of expert system, which e.g. excludes non-logically combinations such as the use of robots and grippers with insufficient load capabilities for the specific use case. In order to fulf interoperability concerns of IT systems, KD$_2$ deals with the creation of consistent data models, improving the scalability and maintainability of the EC. The details and conceptual implementation of each KD are explained in the following subsections.

3.1. KD$_1$ - Business cases (application need and benefit)

Based on business cases designed by the management, the aim of the first knowledge domain (KD$_1$) is to identify, concretise and validate the application scenarios of the aspired EC and the users involved (Fig. 6). Based on this, solution concepts are explored using functional partial prototypes.
Depending on the business objective, both the hardware to be used, e.g. desktop, virtual reality (VR) or augmented reality (AR), as well as the user interface may vary.

<table>
<thead>
<tr>
<th>Engineering configuration based on application- and target group-specific user interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer group (CG) of a specific business case</td>
</tr>
<tr>
<td>Application- and target group-specific user interfaces (UCD-F)</td>
</tr>
<tr>
<td>Knowledge about the specific business cases (KD₁)</td>
</tr>
</tbody>
</table>

Fig. 6. Business cases for EC with different user-centred front-ends (UCD-F)

KD₁ ensures that the economic relevance is addressed first during development before focusing on the concrete technical implementation. The respective user, for example from the sales department of the system integrator or the production planning department of the manufacturing company, should be assisted by the aspired EC in a simple way from the problem to the best possible (automation) solution [13]. The RAS, i.e. the configured result, should be found via a decision chain that is as short as possible [35]. In addition, the usability of the software should be increased to reduce the cost of training employees as well as to minimise the entry barriers for usage.

Based on partial functional prototypes, the business requirements and respective functions can be concretised, evaluated and prioritised iteratively by developing them independently from a more complex larger system. The functions could serve as the development basis for a web-based EC platform into which they will be successively integrated. The functional partial prototypes also create a communication basis for KD₁, as well as to minimise the entry barriers for usage.

3.2. KD₂ - Best Practice (case-specific RAS solution knowledge)

In order to reduce complexity and build on established engineering experience, BP are introduced as a second knowledge domain (KD₂). The advantage of BP is that they can be acquired and processed independently of other knowledge domains. Furthermore, BP can be used apart from EC, i.e. they can be integrated as an example within various documentations, prototypes or software tools [14].

The top-down development of an EC requires a modular, generic architecture of BP, representing an essential data basis. According to the 3-level architecture shown in Fig. 7, the modularisation of BP for RAS can be done on the basis of process modules, components, layout and visual design objects as well as information objects. [14]

The BP takes into account the components used, such as an endeffector, adapter and robot, as well as their arrangement. In addition, the process-relevant positions or movements of the workpieces are taken into account, e.g. via additional information objects (Fig. 7, IFO 1). In doing so, the material flow as well as the process flow of the automation task are also integrated in this static representation. The BP can then be individualised for the respective situation applying logical RAS engineering knowledge (e.g. in the form of constraints) in the sense of a customisation configuration. [14]

3.3. KD₃ - Logical expert knowledge (generic RAS knowledge)

An essential prerequisite for knowledge acquisition based on the division of labour is the object-oriented approach to the preparation of configuration models resp. knowledge. In general, a distinction is made between tabular, instruction-based, rule-based and constraint-based configuration. As can be seen in Fig. 8, constraint-based configuration enables an object-oriented, component-independent representation of EC knowledge. Furthermore, constraint-based EC are best scalable, since the solution can be found automatically in the context of a search problem. In principle, constraints are rules without a direction of derivation, providing the basis for the reasoning algorithm of the EC. Constraints apply to a group of components, using object classes and properties (Fig. 8).

Thereby, the solution is described indirectly. [25,30,36,37]

Fig. 7. Modular 3-level architecture of Best Practices [14]

Fig. 8. Constraint resp. dependency net according to the basic representation from [30,36] extended by own example in the context of robotics [14]
framework such as Tacton (https://tacton.com) or Choco (https://choco-solver.org). The last step addresses the optimisation and continuous further development of the configuration model. Further details regarding modelling and development of logical expertise can be found in [14,37].

3.4. KD4 - Semantically consistent data models (back-end IT knowledge for interoperability)

From the necessity of consistent data exchange between IT solutions for RAS planning (e.g. [38]), KD4 deals with the development of semantically consistent data models ensuring interoperability. In general, the path towards continuous, consistent modelling can be divided into three stages, representing the historical development of technology enablers in an abstracted way (Fig. 10). Similar to a staircase, the higher stages can only be reached by completing the lower stages [32]. The first stage comprises the established communication standards between or within computers and is only listed here for the sake of completeness (cf. OSI model according to ISO 7498). The second level is reached by introducing neutral data formats such as AutomationML (AML) [39] and knowledge representation languages like Web Ontology Language (OWL) [40].

KD3 was demonstrated via several prototypes implemented in the configuration framework Tacton. To validate KD4, an AML model was built, which was also used for back-end data services, among other things. While all KD were initially developed independently of each other, they were finally incorporated into the overall ROBOTOP framework, representing a web platform for the knowledge-based configuration of RAS (Fig. 11).

5. Conclusion and outlook on future research activities

Within this paper, four independent knowledge domains (KD) were introduced which enable an agile, distributed development of user-centred engineering configurators (EC) by separating the concerns. In KD1, the identification of application scenarios, requirements and solution concepts for business cases of the EC with the help of functional partial prototypes is addressed. With KD2, a modular Best Practice architecture and concepts for preparing such case-specific solution knowledge are considered. KD3 deals with development of the generic configuration models and constraints for the EC via logical partial prototypes. For scalability and interoperability reasons, back-end semantics, i.e. semantically consistent data models and schemas, are considered in KD4.
In the context of future developments, the ideas of the EC are to be linked more closely with process-driven approaches (e.g. [41]) in order to combine the advantages of configuration with those of scalable process automation platforms (e.g. [42]). This process-driven approach has already been implemented in a large project at SAP (e.g. [43,44]), among others.

Acknowledgements

The research project ROBOTOP (01MA17009E) was funded by the German Federal Ministry of Economic Affairs and Energy (BMWi). It is part of the technology program “PaICE – Digitale Technologien für die Wirtschaft”.

References