5th Conference on Production Systems and Logistics

Towards A Modular IT-Landscape For Manufacturing Companies: Framework For Holistic Software Modularization

Sebastian Junglas[1], Martin Perau[1], Dino Hardjosuwito[1], Tobias Schröer[1], Wolfgang Boos[1], Günther Schuh[1]

[1] FIR (Institute for Industrial Management) at the RWTH Aachen University, Aachen, Germany

Abstract

Companies in the manufacturing sector are confronted with an increasingly dynamic environment. Thus, corporate processes and, consequently, the supporting IT landscape must change. This need is not yet fully met in the development of information systems. While best-of-breed approaches are available, monolithic systems that no longer meet the manufacturing industry's requirements are still prevalent in practical use. A modular structure of IT landscapes could combine the advantages of individual and standard information systems and meet the need for adaptability. At present, however, there is no established standard for the modular design of IT landscapes in the field of manufacturing companies' information systems. This paper presents different ways of the modular design of IT landscapes and information systems and analyzes their objects of modularization. For this purpose, a systematic literature research is carried out in the subject area of software and modularization. Starting from the V-model as a reference model, a framework for different levels of modularization was developed by identifying that most scientific approaches carry out modularization at the data structure-based and source code-based levels. Only a few sources address the consideration of modularization at the level of the software environment-based and software function-based level. In particular, no domain-specific application of these levels of modularization, e.g., for manufacturing, was identified.

Keywords: Information System; Modularization; IT Landscape; Manufacturing; Literature Review

1. Introduction

Dynamic production environments lead to a need for adaptable processes in manufacturing companies [1]. Adaptable processes require changeable IT systems and the resulting IT landscapes to enable and support such processes [2]. In this context, an IT landscape is to be understood as the sum of the individual information systems used [3]. Today's major information systems in manufacturing companies are mainly monolithic, rigid, and unsuitable for such a requirement [4]. A typical life cycle of such a monolithic information system lasts 7-10 years, and the information systems' configuration and design are based on the initial requirements from the beginning of the respective life cycle [5]. The needed incremental adaptation of these changing requirements over time only happens individually and is not structured, and thus the information system cannot adapt to the dynamic environment.

Studies show that future-proof information systems and the resulting IT landscapes must be flexible, functional, and consistent [6]. This can be achieved through a modular approach. A modular approach offers the added value that individual modules can be developed independently, used selectively, and configured according to demand [7]. Furthermore, due to the above-mentioned requirement for flexibility, a quick implementation can be carried out in the event of changing requirements [7]. Modularization of information systems is a promising path to a holistic best-of-breed approach [8,9]. A best-of-breed approach requires that the best solution for the specific use case is selected and implemented (cherry-picking) [10]. Modularizing

DOI:
ISSN: 2701-6277

information systems can enable the suitable composition of modules for the needed functionality fit within a modular IT landscape. The functionality is tailored to the respective use case, as just the fitting modules (in the sense of the best-of-breed approach) can be used. This suitable functionality can be achieved in modules, if modules consist of business-related functions that fit together and are only loosely connected to their environment. Hereby a module is defined as loosely coupled with other modules and having a high cohesion of its components within the module [12]. This definition of a module also has an advantage for consistency - interfaces will only exist where they are needed. This is in direct contrast to IT landscapes as they exist today. In today's IT landscapes, monolithic information systems prevail. Their size and complexity makes them slow to change and hinder the capacity of companies to adapt to a changing environment [11]. A monolithic system always means compromises. Thus, it is crucial to focus on the modularization of information systems to enable modular IT landscapes and adaptable processes.

This paper analyzes and presents existing approaches to modularizing information systems within IT landscapes. Particularly relevant is the classification in a domain-specific reference. The aim is to present distinct types, respectively levels, of software modularization and to analyze whether any types of software modularization are underrepresented in the scientific consideration to date. This will test the hypothesis that domain-specific modularization of information systems in manufacturing companies does not exist in the necessary form seen in the market. This goal is pursued with systematic literature research. The research question to be answered is: What types of software modularization exist, and are any particular types of software modularization underrepresented in science?

This paper is divided into five chapters. After the introduction, the conducted method is discussed and presented in detail. Chapter 3 presents the results of the research. Chapter 4 deals with the discussion of the results obtained before a critical appraisal and an outlook is given in Chapter 5.

2. Applied methodology

The overarching method is based on the systematic literature review framework according to vom Brocke et al. and consists of five steps [13]:

1. Definition of review scope
2. Conceptualization of topic
3. Literature search
4. Literature analysis and synthesis
5. Research agenda

Defining the scope of the literature review ensures that the objectives of the given research question are systematically implemented throughout the literature review process. The taxonomy of the literature review, according to Cooper, supports this [14]. The figure below shows our definition of the review scope for our research question.

Characteristic	Categories			
goal	integration	criticism	central issues	
perspective	neutral representation	espousal of position		
organisation	historical	conceptual	methodological	
audience	specialised scholars	general scholars	practitioners / politicians	general public
focus	research outcomes	research methods	theories	applications
coverage	exhaustive	exhaustive and selective	representative	central/pivotal

Scope of this paper   None scope of this paper

Figure 1: Definition of the review scope of this paper (Following Cooper [14])

The goal of the paper is the integration or, more specifically, the synthesis of existing approaches to software modularization and the derivation of existing central issues of these approaches for the application context of the design of the modular IT landscape of manufacturing companies. It follows that the literature review is classified as an espousal of position, as we want to synthesize the literature accordingly to consider whether the consideration of the application of software modularization approaches has already been studied scientifically. A view from an espousal perspective does not contradict the necessary scientific neutrality as long as the conclusions are logical and transparent, which we ensure through the detailed description of our methodology [14]. The organization of the relevant sources to generate the knowledge to answer the research question is accomplished through the development of a framework. The literature review is conceptual in character and aims at combining similar approaches [14], in our case, software modularization approaches. The framework is aimed at scientists from different fields, as the paper's topic is located at the interface between computer science and production management. The focus of the material of the literature review cannot be classified precisely, as existing research methods, research outcomes, applications, and theories can contribute to the development of the framework. According to Cooper, exhaustive and selective coverage is suitable for an integrative literature review [14], as it allows for the necessary comprehensive investigation with the simultaneous synthesis of the literature. Accordingly, we have chosen the exhaustive and selective coverage category for our literature review. The definition of the review scope serves as a guideline for the entire literature review and as input for the next steps.

The second step is the rough conception of the topic field, which enables the derivation of the search strings for the literature review steps [13]. The rough conception of the topic field is based on the research questions. The overall theme of the paper are modular IT landscapes, and the research questions focus on the formation of modules within information systems (and thus IT landscapes) under the application of a method. Modules in this paper's context are formed from IT landscape information systems. To achieve this, it is necessary to design information systems modularly. Thus, in the literature search, modules of information systems are focused. In the context of this paper, modules are understood as system components that should have a loose coupling between each other and a high cohesion of their respective internal elements [12,15,16]. Furthermore, the term “software” is synonymous with “information system” in the literature research. For the procedure of module formation, different terms are used. The term “modularization” describes the division of an information system into modules [15]. Other terms used by authors instead of modularization in this context are “structuring” and “decomposing”[15,16], which are used as synonyms for “modularization” in the literature review. The keywords for the systematic literature research are derived from the formation of modules within information systems under the application of a method.

For the search strings of the systematic literature research, the term “module” is used as keyword 1. Furthermore, to ensure that the modules considered in the literature search, are modules of information systems, the term “information system” is used as keyword 2, as well as its synonym “software”. Since the research questions are about forming modules and not just examining modules themselves, the method of forming modules is focused on in the literature review. Thus, the term “modularization” as well as its synonyms “structuring” and “decomposing” are used as keyword 3. Based on this conception of the thematic field, the following search strings were developed (see Table 1).

Table 1: Search strings for the literature review

Nr.	Key Word 1	Boolean	Key Word 2	Boolean	Key Word 3
1	Module	AND	Software	AND	Structuring
2				AND	Modularization
3				AND	Decomposing
4		AND	Information system	AND	Structuring
5				AND	Modularization
6				AND	Decomposing

In the third step, a literature search was conducted. Based on a search string search in electronic databases, a multi-stage filtering and screening process was carried out following vom Brocke et al. and Moher et al. to reduce the number of publications to be considered systematically [13,17]. The framework of this process was documented using the STARLITE method (see Table 2).

Table 2: Documented literature search according to STARLITE [18]

Element	Description
Sampling strategy	Consideration of all literature within defined boundaries
Type of studies	Scientific articles, books, and PhD-Thesis
Approaches	Search strings search in electronic databases
Rang of years	2014 until 06/2023
Limits	No duplicates; only English or German publications; publicly available publications
Inclusion and exclusion	Including: software modularization or information system modularization; Excluding: Product modularization; Ontology modularization; service modularization
Terms used	See Table 1.
Electronic sources	Scopus

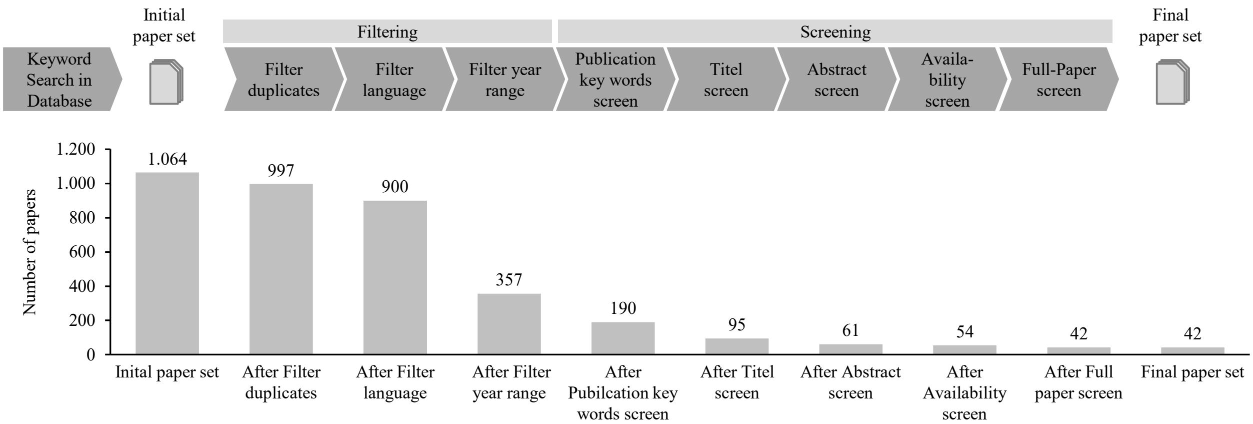

Figure 2: Filtering and Screening process of the literature search

The resulting publications, including the documentation of the filtering and screening process, can be found in the following document: https://epub.fir.de/frontdoor/index/index/docId/2704.

The literature review and synthesis are to be conducted to identify the approaches used for software modularization and, using these approaches, to derive which levels of software modularization. For this purpose, the objects of modularization used per paper are to be first analyzed and, based on these objects, a framework of the different levels of software modularization is to be developed by assigning thematically related objects to a level of software modularization. Objects of modularization describe those objects that are combined into modules by the respective modularization process. The derivation of the research agenda is done in section 4 based on the findings of section 3.

3. Results of the literature review

This chapter presents the results of the literature review. The research papers were examined according to different approaches to modularization – concrete the objects of modularization. Objects of modularization refer to those aspects that are combined and thus result in a module. Overall, the following objects of modularization were identified during the described literature review: requirements, organizational units, conceptual model use case, GUI-based functions, artifacts, classes, subset of program functions, features, domain entities, aspects, packages, and source code. The identification of the mentioned objects of modularization within the examined papers shows that different kinds of objects are considered. This becomes clear for example since requirements determine the needs for a software application, classes describe the software-technical structuring of methods and source code the executable syntax of the software application.

Since the paper's goal is to synthesize existing approaches to software modularization and derivate existing central research gaps in the application context of the design of the modular IT landscape of manufacturing companies, an initial, generally valid approach for the synthesis is necessary. The V-Model is used as the basis of the synthesis, enabling a generic and holistic view of the software development process. The V-Modell is an established standard in software development and enables the decomposition of software development processes into different levels of consideration (see Figure 3) [19–21].

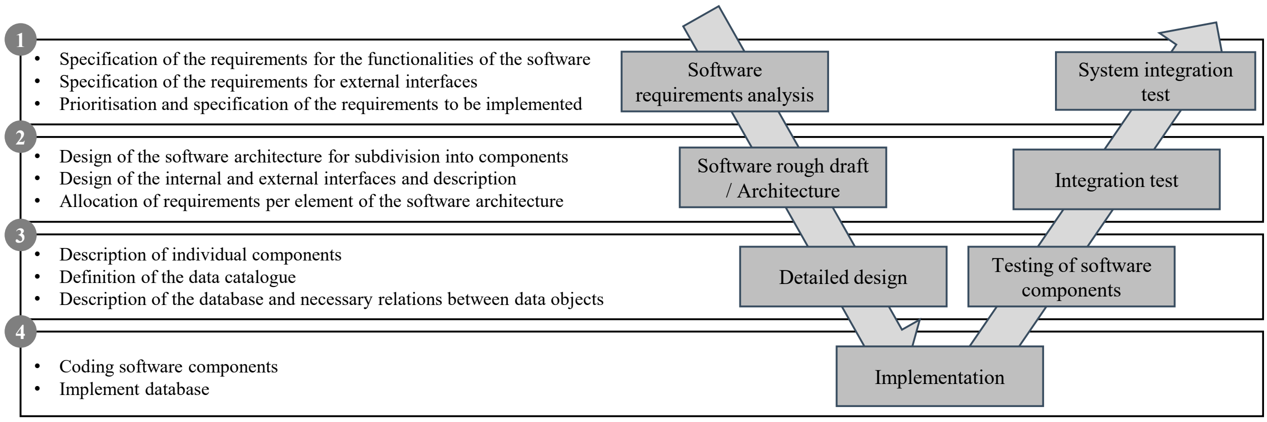

Figure 3 V-Modell for Software Engineering following Dröschel et al. [19]

A strongly business-driven approach characterizes the first step of the V-Modell. The software requirement analysis specifies requirements for software functionalities and external interfaces. Then, in the second step, the software is divided into individual components, and the interfaces between the individual components are defined. A critical component is also the assignment of requirements to the respective components. In the third step of the V-Modell, the data-related description of individual software components takes place through component descriptions, data catalogues, and the necessary data relationships. The fourth step of the V-Modell considers the implementation of the data-related description of the individual components through the implementation of the components through source code. [19] The subsequent consideration of the different test steps is not relevant in the context of this paper and will thus not be discussed.

A framework for holistic software modularization was developed based on the V-Modell. Four levels of software modularization exist: software environment-based, software function-based, data structure-based, and source code-based (see Figure 4). The levels differ in their weighting regarding the focus on business-specific aspects towards a focus on technical aspects with the given order. This differentiation results from the associated objects of software modularization. In addition, the mentioned levels of software modularization are hierarchically structured. This means that a formed module in a level of software modularization can be further modularized on a lower level. The modularization on the lower level uses different objects than within the higher modularization level.

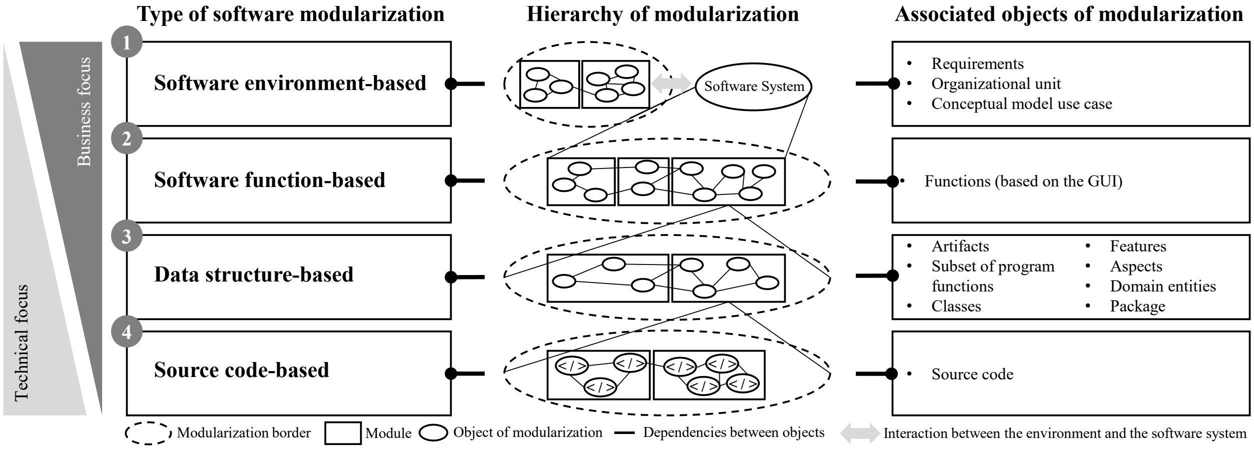

Figure 4 Framework for holistic software modularization

The most business-focused view of software modularization is the software environment-based modularization, in which primarily external objects are considered as the modularization objects, e.g. business units. The level of software function-based modularization considers the decomposition of the software system into individual components, considering the functions of the entire system. The level of data structure-based modularization has a strong technical focus. Here, the primary goal is to optimize the data structure of a single software component by modularizing various objects, e.g., classes. The fourth level of software modularization, source code-based modularization, has an exclusively technical focus. In the following, the respective levels of modularization are described.

Software environment-based modularization (Level 1)

At this level, the software environment is included in the modularization process. The software system is influenced by various actors and activities in the environment. These factors can be used for modularization approaches. The strongest business focus can be seen at this level of modularization. No technical factors are included in the modularization efforts. The modularization objects used at this level are requirements, organizational units, and conceptual models (Use Case).

Requirements (e.g. in [22]) refer to the functional needs of software [12]. Requirements are combined in logical groups and modules are developed from them [22]. Another object of modularization are organizational units (e.g. in [23]). Here, modules are formed that can be designed based on the organizational unit in which they are used (e.g. production planning, sales, or procurement). Organizational units are structures in a company, that can be complemented by the responsibilities of the unit and size means the number of involved people [24]. The object of modularization is therefore directly dependent on the company's structure. Use cases refer to fields of application that are mapped in software. The Use Case describes the interaction between a product and the actor [25]. Based on the definition of use cases, various specific situations can be mapped, which can then be combined in software as a module.

Software function-based modularization (Level 2)

At the level of software function-based modularization, the individual necessary components are used at the function level to define and design suitable modules. This is the first time that a technological design of business components has taken place. The analysis of the available research papers just shows one approach where functions based on the GUI elements (e.g. main window, menu bar) are used as an object of modularization [26].

Data structure-based modularization (Level 3)

At the level of data structure-based modularization, mainly technical components are used to form modules. There is a strong data-related focus that structures the individual data points and makes them available as objects for modularization. The research conducted here shows a large number of objects that are used to form modules. These are artifacts, subsets of program functions, classes, features, aspects, domain entities, and packages.

Artifacts (e.g. in [27]) are software elements produced during the software development process. In this context, an artefact can be defined as the structure, routines, or values of software [28]. Subset of program functions (e.g. in [29]) means those functions that are necessary for the operation of the software. This can mean, for example, memory capacities. The object of modularization here therefore refers to the higher-level technical functions and not to the procedural implementation of tasks. Domain Entities (e.g. in [30]) are objects in software always related to a data point. This means that a recurring data point classifies as an entity and thus works like a variable in software. The reverse conclusion is that the object of modularization tends to lead to very finely granular modules. A package combines various individual classes of software. This shows the connection to the classes that are also analyzed as objects of modularization (e.g. in [31]). Classes unite features of the software. Features are, for example, attributes or methods. These classes then have the same features. [12] Here a hierarchical arrangement of the objects of modularization can be seen, which results in a different granularity of the modules depending on the application. Aspects of software (e.g. in [23]), on the other hand, refer to software functionalities that do not contribute to the business logic, such as the logging of input data. In summary, modularization is carried out at all levels of a software hierarchy. The objects of modularization are usually sub-objects of a higher hierarchy level.

Source code-based modularization (Level 4)

On the fourth level of modularization, only technical objects are used to form modules. The reference to the business logic is no longer present and a purely technical implementation is carried out or used for modularization. Only one modularization object can be assigned to this level - the source code.

Source code (e.g. in [22]) can be defined as a human-readable form of a program in a programming language. Modularizations at the code level mean the sensible combination of areas of the source code. Here, modularization means that lines of code are clustered which can be expected to facilitate the implementation or execution of the software [32]. This makes the developer's work easier, as no things must be implemented twice, which also improves the performance of the software.

4. Discussion of the results

The results of the systematic literature research clearly show that there is a focus area in software development in which modularization activities are already taking place today. In the big picture, modularization as a driver for making software more flexible and improving its functionality aims to break down software boundaries and thus design entire IT system landscapes through modules. The papers analyzed here clearly show that the efforts toward modularization are only considered in a detached IT system. The reference to the structure of an IT system landscape is completely missing. Furthermore, in the division of the classification of the papers in the developed framework, it becomes clear that only a few objects of modularization could be identified on the level of software environmental-based and software function-based modularization (level 1 and 2) (see Figure 6).

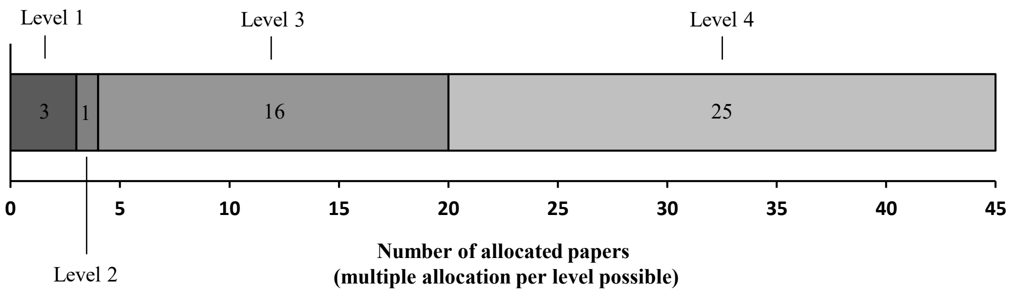

Figure 5 Distribution of the papers in the levels of the developed framework

Modularization takes place on the data structure-based and software code-based modularization level (levels 3 and 4), but the reference to the actual business logic is mostly missing. On the first two levels, it is about requirements and the realization in the respective business logic - there are hardly any objects of modularization in use in science. Especially in this context, the domain reference is absent, and the initial established hypothesis established can be verified. Domain reference and the consideration of requirements or functions of business logic are not considered. However, this domain-specific perspective is necessary to form modules considering the business logic. Thus, the research questions can be answered. Predominant modularization approaches take place, for the most part, on the technical-focused levels of software modularization. This means that mainly software development components are considered that have no connection to business logic. The reference to the application domain could not be identified. This shows the research gap in business-focused modularization.

5. Conclusion and Outlook

In theory, modular IT landscapes and the modular information systems they require have many advantages, such as rapid adaptability and transparent interfaces, while at the same time ensuring comparatively low costs. However, implementing holistic, modular IT landscapes has not yet been implemented in practice. Our literature analysis shows that there is a lack of scientific foundations for software modularization within the higher level of software development or the development of IT landscapes. Based on systematic literature research, objects of software modularization were examined in existing scientific approaches, and a framework for holistic software modularization was derived from this. Types of software modularization can be differentiated according to their weighting between business focus and technical focus. The main business focus is the modularization type of software environmental-based modularization followed by software functional-based modularization. The types of data structured-based and source code-based modularization have a technical focus. Modularization on the levels of software environmental-based or functional-based modularization is only considered by a minority of authors. This lack of modularization approaches at these levels shows the need for future research. For a holistic modularization of software and the implementation of modular IT landscapes, modularization approaches are needed at these levels of software development. Due to the high relevance of business logic within these levels, a generic view of modularization is insufficient. It requires the consideration of business for the respective domain-specific application areas. The resulting modularization approaches on the levels of software environmental-based and software functional-based modularization are building blocks for future-proof IT landscapes.

Acknowledgments

The underlying research project Serial Production of Electrolyzers in the Gigawatt Range (SEGIWA) is funded by the German Federal Ministry of Education and Research (BMBF) as part of the joint project H2Giga – funding code: 03HY121K.; Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC-2023 Internet of Production – 390621612

References

[1] Bischoff, J., Taphorn, C., Wolter, D., Braun, N., Fellbaum, M., Goloverov, A., Ludwig, S., Hegmanns, T., Prasse, C., Henke, M., Hompel, M. ten, Döbbeler, F., Fuss, E., Kirsch, C., Mättig, B., Braun, S., Guth, M., Kaspers, M., Scheffler, D., 2015. Erschließen der Potenziale der Anwendung von ‚Industrie 4.0' im Mittelstand: Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi), 401 pp.

[2] Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N., Brieke, M., 2007. Changeable Manufacturing - Classification, Design and Operation. CIRP Annals 56 (2), 783–809.

[3] Hanschke, I., 2010. Strategic IT Management. Springer Berlin Heidelberg, Berlin, Heidelberg.

[4] Ditsche, J., Rohrhirsch, D., Grunow, S., Fritze, R., 2022. The digital dilemma: Why companies struggle to master digital transformation, 16 pp.

[5] Schneider, S., Wollersheim, J., Krcmar, H., Sunyaev, A., 2018. How do Requirements Evolve over Time? A Case Study Investigating the Role of Context and Experiences in the Evolution of Enterprise Software Requirements. Journal of Information Technology 33 (2), 151–170.

[6] Gronau N., 2019. Trend im ERP-Markt 2019: ERP Technologie. https://erp-management.de/themen/erp-technologie/artikel/trends-im-erp-markt-2019/. Accessed 4 August 2023.

[7] Böhmann, T., 2004. Modularisierung von IT-Dienstleistungen: Eine Methode für das Service Engineering. Deutscher Universitätsverlag, Wiesbaden, 317 pp.

[8] Kletti, J. (Ed.), 2015. MES - Manufacturing Execution System: Moderne Informationstechnologie unterstützt die Wertschöpfung, 2. Auflage ed. Springer Vieweg, Berlin, Heidelberg, 288 pp.

[9] Rizni, M.I., Poravi, G., 2018. Best of Breed ERP: A Dashboard for Strategic Decision Makers. International Conference of Intelligent Systems, Modelling and Simulation 8, 58–61.

[10] Hanschke, I., 2010. Strategic IT Management. Springer Berlin Heidelberg, Berlin, Heidelberg, 348 pp.

[11] Newman, S., 2015. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Inc.

[12] Balzert, H., 2009. Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Auflage ed. Spektrum Akademischer Verlag, Heidelberg, 624 pp.

[13] Brocke, J. Vom, Cleven, A., Niehaves, B., Plattfaut, R., Riemer, K., Simons, A., 2009. Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process, in: Information systems in a globalising world. Challenges, ethics and practices. 17th European Conference on Information Systems (ECIS) 2009, Verona. 08.-10.06.2009, pp. 2206–2217.

[14] Cooper, H.M., 1988. Organizing Knowledge Syntheses: A Taxonomy of Literature Reviews. Knowledge in Society 1 (104), 104–126.

[15] Parnas, D.L., 1972. On the criteria to be used in decomposing systems into modules. Communications of the ACM 15 (12), 1053–1058.

[16] Yourdon, E., Constantine, L.L., 1979. Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Prentice-Hall, River, NJ, United States, 473 pp.

[17] Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. BMJ (Clinical research ed.) (339), 332-336.

[18] Booth, A., 2006. ''Brimful of STARLITE'': toward standards for reporting literature searches. Journal of the Medical Library Association JMLA 4 (94), 421–430.

[19] Dröschel, W., Wiemers, M., 1999. Das V-Modell 97: Der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz. De Gruyter Oldenbourg, München, 696 pp.

[20] Gräßler, I., Oleff, C., 2022. Systems engineering: Verstehen und industriell umsetzen. Springer Vieweg, Berlin, Heidelberg, 235 pp.

[21] Rausch, A., Broy, M., 2008. Die V-Modell XT Grundlagen, in: Höhn, R., Höppner, S. (Eds.), Das V-Modell XT. Anwendungen, Werkzeuge, Standards. Springer, Berlin, Heidelberg, New York, pp. 1–28.

[22] Naseem, R., Deris, M.B.M., Maqbool, O., 2014. Software modularization using Combination of Multiple Clustering, in: 17th IEEE International Multi Topic Conference 2014, Karachi, Pakistan. 08.12.2014 - 10.12.2014. IEEE, pp. 277–281.

[23] Berta, P., Vranic, V., 2019. Synergy of Organizational Patterns and Aspect-Oriented Programming, in: 15th International Scientific Conference on Informatics, Poprad, Slovakia. 20-22 November 2019. IEEE, pp. 439–444.

[24] Petersen, K., Wohlin, C., 2009. Context in industrial software engineering research, in: 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM), Lake Buena Vista, FL, USA. 15.10.2009 - 16.10.2009. IEEE, pp. 401–404.

[25]Schach, S.R., 2011. Object-oriented and classical software engineering, 8th ed. ed. McGraw-Hill, New York, 667 pp.
[26]Seo, Y.-S., Huh, J.-H., 2022. GUI-based software modularization through module clustering in edge computing based IoT environments. J Ambient Intell Human Comput 13 (3), 1625–1639.
[27]Tabrizi, A.H.F., Izadkhah, H., 2019. Software Modularization by Combining Genetic and Hierarchical Algorithms, in: 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran. 28.02.19 - 01.03.19. IEEE, pp. 454–459.
[28]Benbasat, I., Zmud, R.W., 2003. The Identity Crisis within the Is Discipline: Defining and Communicating the Discipline's Core Properties. MIS Quarterly 27 (2), 183–194.
[29]Riesco, A., Zhang, M. (Eds.), 2022. Formal Methods and Software Engineering, 1st ed. 2022 ed. Springer International Publishing; Imprint Springer, Cham, 440 pp.
[30]Gonçalves, N., Faustino, D., Silva, A.R., Portela, M., 2021. Monolith Modularization Towards Microservices: Refactoring and Performance Trade-Offs, in: 18th International Conference on Software Architecture Compansion (ICSA-C), Stuttgart, Germany. 22-26 March 2021. IEEE, pp. 54–61.
[31]Paixao, M., Harman, M., Zhang, Y., 2015. Multi-objective Module Clustering for Kate, in: Search-Based Software Engineering. 7th International Symposium, SSBSE, Bergamo, Italy. 5-7 September, 2015. Springer International Publishing, Cham, pp. 282–288.
[32]Jiaze, S., Xu, Y., Wang, S., 2018. PSO with Reverse Edge for Multi-Objective Software Module Clustering. International Journal of Performability Engineering 14 (10), 2423–2431.

Biography

Sebastian Junglas (*1991) has been working at the Institute for Industrial Management (FIR) at the RWTH Aachen University since 2020, first as a project manager and since 2022 as group lead production control. He studied mechanical engineering and management at the RWTH Aachen University and Production Management at the Technical University Wroclaw, Poland. In his research, he focuses on the modularization of IT landscapes, especially in the area of Production-IT.

Martin Perau (*1996) studied production engineering and management, business, and economics at the RWTH Aachen University and in Finland at the Aalto University in Helsinki. He is a research assistant at the Institute for Industrial Management (FIR) at RWTH Aachen University in the Department of Production Management since 2022. His research focuses on IT landscapes and information systems and their potential to support sustainable and circular production processes.

Dino Hardjosuwito (*1988) has been working at the Institute for Industrial Management (FIR) at the RWTH Aachen University since 2020, first as a project manager and since 2021 as a group lead production planning. He studied mechanical engineering as well as production engineering at the RWTH Aachen University and also studied at Carnegie Mellon University in Pittsburgh, USA, and ETH Zurich in Switzerland. In his research, he focuses on the modularization of IT-landscapes especially in the area of business process management.

Tobias Schröer (*1991) has been working at the Institute for Industrial Management (FIR) at the RWTH Aachen since 2016, first as a project manager and since 2020 as head of the Department Production Management. As such he leads a variety of applied research and consulting projects. In his research, he focuses on the real-world applications of business software for operations in production and logistics.

Prof. Dr.-Ing. Wolfgang Boos, MBA (*1975) studied mechanical engineering at RWTH Aachen University after training as a tool mechanic in the field of stamping and forming technology. Following his studies, he completed his doctorate at the Machine Tool Laboratory (WZL) with distinction and became the managing senior engineer of the Chair of Production Systems at the WZL. Since 2023 he is the managing director of the Institute for Industrial Management (FIR) at the RWTH Aachen University.

Prof. Dr.-Ing. Dipl.-Wirt.-Ing. Günther Schuh (*1958) is the director of the Institute for Industrial Management (FIR) at RWTH Aachen University, holds the Chair for Production Systems at the WZL, and is a member of the board of directors of the Fraunhofer Institute for Production Technology IPT in Aachen. Professor Schuh is a member of several supervisory boards and boards of directors.