Modular maintenance instructions architecture (MMIA)

Kristoffer Vandrup Sigsgaard, Julie Krogh Agergaard,
Niels Henrik Mortensen, Kasper Barslund Hansen and Jingrui Ge
Department of Civil and Mechanical Engineering, Technical University of Denmark, Lyngby, Denmark

Received 12 August 2021
Revised 4 May 2022
14 July 2022
Accepted 21 August 2022

Abstract

Purpose – The study consists of a literature study and a case study. The need for a method via which to handle instruction complexity was identified in both studies. The proposed method was developed based on methods from the literature and experience from the case company.

Design/methodology/approach – The purpose of the study presented in this paper is to investigate how linking different maintenance domains in a modular maintenance instruction architecture can help reduce the complexity of maintenance instructions.

Findings – The proposed method combines knowledge from the operational and physical domains to reduce the number of instruction task variants. In a case study, the number of instruction task modules was reduced from 224 to 20, covering 83% of the maintenance performed on emergency shutdown valves.

Originality/value – The study showed that the other methods proposed within the body of maintenance literature mainly focus on the development of modular instructions, without the reduction of complexity and non-value-adding variation observed in the product architecture literature.

Keywords Inspection, Maintenance planning, Quality control, Maintenance analytics

Paper type Research paper

Introduction

With increasing production complexity comes increasing maintenance complexity. As facilities grow in size and production volume, ensuring that the right maintenance takes place at the right time becomes difficult (Agergaard *et al.*, 2021; Sigsgaard *et al.*, 2021a, b). The instructions describing the actions to take during maintenance are essential in the maintenance process. Low-quality instructions lead to low-quality maintenance work, as well as extra idle time spent understanding the instructions. When instructions are written in a free-text format for individual pieces of equipment, the amount of variation becomes large in production plants with hundreds of thousands of pieces of equipment. When the variation becomes too large, it becomes difficult and time consuming to evaluate the actions taken during maintenance, making it difficult to make decisions about how maintenance should be performed in the future (Agergaard *et al.*, 2021).

The study presented in this paper was performed in a case company that has seen increasing complexity and variance in its maintenance instructions. The operational portfolio contains many instructions with little to no variation in effect but many variants in formulation across two languages. This increases the time it takes to formulate and understand the maintenance instructions. The complexity has had a negative effect on the

---

The authors would like to thank DTU Offshore for the funding of this project.

Journal of Quality in Maintenance Engineering
Vol. 29 No. 5, 2023
pp. 50-67
Emerald Publishing Limited
1355-2511
DOI 10.1108/JQME-08-2021-0063

quality of the maintenance and the time spent on idle tasks, such as reading the instructions. The complexity of the instructions has also made it difficult to evaluate the current maintenance situation because the amount of variation is so large that the maintenance is no longer comparable.

Several studies have shown that more precise and consistent job instructions enable less skilled personnel to carry out maintenance tasks that would otherwise require more experienced technicians (Harris, 1994). Within the field of maintenance instructions, several techniques for standardizing instructions have been proposed, including the task-oriented adaptive maintenance system (TOAMS) (Huang *et al.*, 2015) and customized maintenance documents (Huang *et al.*, 2014). These methods aim to customize maintenance descriptions based on the particular tasks at hand and the end-user experience. Furthermore, Toscano (2000) investigates the use of interactive electronic technical manuals (IETMs), which seek to provide users with just-in-time instructions for maintenance tasks. These methods seek to modularize maintenance instructions and make them configurable to individual maintenance job requirements. However, the methods do not consider whether the variation found in the maintenance instructions is value-adding or non-value-adding. Because unnecessary modules within a configurable architecture are complex and time-consuming to maintain, this paper proposes the Modular Maintenance Instructions Architecture (MMIA) as a method for evaluating and formulating a maintenance instruction architecture. The method uses knowledge from both the operational and physical domains to evaluate the value addition of maintenance instruction variants.

The MMIA was created for multicomponent systems, while the methods identified in the literature are limited to single-component maintenance, making the method more suitable for large production facilities. When creating modules for multicomponent systems, other dimensions play a role in the decision-making because they must differentiate between action differences and differences in the physical and process dimensions. Differences in the physical dimension occur when there are different types of equipment that have variable requirements in terms of maintenance. Differences in the process dimension occur when multiple pieces of equipment are being maintained at different stages at any given time (Sigsgaard *et al.*, 2021a). The study presented in this paper was shaped by the following research question:

How can linking various maintenance dimensions help decrease the complexity of maintenance instructions?

This paper first presents the approach to the research question. A literature review then highlights methods from the maintenance and instruction digitalisation literature. The literature review is supported by the methods of value-addition analysis, derived from product architecture theory, as an addition to the maintenance and instruction literature. The method developed in collaboration with the case company is then highlighted. Finally, a case study using the proposed MMIA method is presented. The study evaluates real maintenance instruction data for a set of safety-critical valves.

Research approach

The research question was approached with the design research methodology (DRM) (Blessing and Chakrabarti, 2009). The need for a method for the evaluation of the value-addition of variety was observed in the case company. The company had lost the overview of the variation in the maintenance instructions and was struggling to make decisions about changes on a larger scale. Approaches to the standardization of maintenance instructions were identified in the literature, but the identified studies did not take the value addition of the variants into consideration. Instead, approaches taken from product and service modularization led to the conceptualization of the proposed method, the Modular Maintenance Instruction Architecture (MMIA). The method was applied in the case company to further iterate and test its applicability. The development of the proposed MMIA and the case study were performed over a six-month period. The case company is a major production company that operates major, continuous production facilities. To limit the scope of this initial study, the case study focused on a systematic analysis of 231 variants of maintenance instructions that describe the preventive maintenance planned for 1,941 safety-critical valves.

Knowledge about the company's maintenance process was collected through internal documents and the company's Computerized Maintenance Management System (CMMS). Information on the maintenance jobs performed on the safety-critical valves was also collected from the CMMS. The data included all maintenance performed over a five-year period. Information about the maintenance jobs included the number of hours, the dates of the performed maintenance, and the maintenance instructions. The physical characteristics and locations of the valves were also extracted.

The findings from the case study were validated through workshops, meetings, and semi-structured interviews with internal maintenance experts, including maintenance workers, maintenance responsibles, system responsibles, and others. The response and feedback received from these key company figures has been vital to the validation of the results of the analysis and the assessment of the developed model.

The study presented in this paper was performed during an MSc project and a BSc project at the Technical University of Denmark, Department of Mechanical Engineering, Section of Engineering Design and Product Development.

Literature review

This section introduces literature on maintenance instructions and the digitalization of documents. The identified methods on maintenance instructions and the digitalization of documents fail to account for the complexity and non-value-adding variety involved in the instructions used for maintenance in large production companies. Product architecture methods have successfully been applied to handle complexity and identify non-value-adding variety. The review is therefore further supported by the product architecture literature.

Maintenance instructions

This section introduces the literature on maintenance instructions. Maintenance instructions are the descriptions of the actions to be performed during the maintenance. As such, the literature in this section describes the action dimension of maintenance (Sigsgaard *et al.*, 2021a).

Maintenance instructions, plans, operations, or tasks are descriptions of the maintenance to be performed (Dansk Standard, 2016). This paper uses the term "maintenance instructions" to refer to a collection of set tasks for completing a maintenance goal. Instructions can be non-knowledge-based or knowledge-based. Non-knowledge-based tasks describe details of the action to be performed so that the person following the instructions does not need any prior knowledge (Jacobs, 2017). A non-knowledge-based instruction might read, "Take a sample from the oil using the pipette in the test kit. Put sample into test liquid and wait 5 min for reaction. Note the amount of water in oil using the color scale from the test kit." Knowledge-based instructions require the person performing the task to have knowledge of the task at hand outside of the instructions (Jacobs, 2017). An example of a knowledge-based task might read, "Measure water in oil."

Paper-based instructions have, until recent years, been the cheapest and easiest way to achieve portable instructions. However, the introduction of tablets and smartphones has changed this significantly. In most major companies, an increasing share of the internal documentation is being digitalized. One example is the US Army, which converted 17,000 pieces of paper to digital files as a part of its Army Digitization Program (Toscano, 2000). As a result of this digitalization, a large amount of information is available in a new way. This can allow for the communication of the instructions in more detailed formats, such as animations and videos, which can improve user understanding (Pham *et al.*, 2000). Several methods have been proposed for the digitalization of documents. The following paragraphs highlight a selection of methods used to digitalize manuals and instructions.

In maintenance, the development of interactive electronic technical manuals (IETM) is a result of the industry moving away from paper-based, "a to z" documents and toward digitalized versions that ensure that the correct instructions are presented at the right time. Introducing this format of instructions improves the instructions, making it possible to involve personnel with fewer skills. This indicates that non-knowledge-based instruction is more easily achieved using this type of format. The digitalized instructions then function as a means of reducing the demand for experienced technicians within maintenance departments (Pitblado, 1991). Similar results are seen in newer studies on the application of augmented reality (AR) in maintenance instructions. When introducing AR solutions instead of paper- or PDF-based methods, the amount of time spent on maintenance, as well as the number of errors, is reduced (Fiorentino *et al.*, 2014; Havard *et al.*, 2021; Mourtzis *et al.*, 2020).

Pham *et al.* (2000) suggest a type of knowledge-based manual that allows a system to decide what instructions are needed based on user inputs. In comparison to paper-based instructions, in which all necessary and unnecessary steps will need to be shown, only the necessary information is included for each individual case, without having to increase the number of instructions. This type of system can provide tailored solutions that fit the level of expertise of the technician (Huang *et al.*, 2014).

Horn (1993) discusses the use of the theory of structured writing to facilitate the change from sequential, printed paper-based instructions to chunks of information configured by a computer system. Structured writing is defined as "a precise modular concept ('information blocks') that are firmly grounded in a taxonomy of information types" (Horn, 1993, p. 4). Information blocks are basic units that replace a paragraph and contain text and graphs.

Setchi *et al.* (2006) outline the methodology of "intelligent product manuals" (IPMs), which are an intelligent way to show consumer-product user manuals to end-users. Primarily, IPMs utilize the Internet and rests on the same idea of applying expert knowledge, product life-cycle information, and hypermedia to provide just-in-time support (Pham *et al.*, 2000).

The methods introduced in this section focus on how to represent and convert the structure of the instructions into digital solutions. However, none of the identified methods evaluate whether the amount of variation within the operations is value-adding. The following section explains how this was achieved in product and service architecture research.

Product architecture

Performing development product by product leads to many products that have overlapping attributes. Having many products increases complexity, but with their overlapping attributes, this complexity tends not to provide value (Meyer and Lehnerd, 1997; Wilson and Perumal, 2009). However, variance is required to be competitive, making a trade-off occur between supplying a large amount of variety and not introducing large production costs (Simpson *et al.*, 2014). Modularized product architectures offer a solution to such increased portfolio complexity (Meyer and Utterback, 1992; Mortensen *et al.*, 2019; Otto *et al.*, 2016) and have been successfully adopted in many companies over the last three or four decades (Meyer and Lehnerd, 1997; Wilson and Perumal, 2009). More recent years has seen an introduction of product architecture and modularization approaches in service portfolio management (de Blok *et al.*, 2014; de Mattos *et al.*, 2021; Eissens-van der Laan *et al.*, 2016; Johnson *et al.*, 2021; Løkkegaard *et al.*, 2016). Likewise, the study by Sigsgaard *et al.* (2021a) showed promising results on the part of an initial step into the application of product architecture approaches in a maintenance management context. Services and maintenance are similar in the sense that they are operational and provide intangible deliverables that provide value. However, where services are tailored to be delivered to a customer, the recipient of value in maintenance is the asset or production owner that requires safe, continuous production (Sigsgaard *et al.*, 2021a). This section introduces concepts from the product, service, and maintenance architecture and modularization literature to add to the application of architectures and modularization in maintenance.

Product architecture is a widely studied subject that has a number of definitions. One widely accepted definition of the product architecture was given by Ulrich (1995): "(1) the arrangement of functional elements; (2) the mapping from functional elements to physical components; (3) the specifications of the interfaces among interacting physical components" (Ulrich, 1995, p. 420). As such, a product architecture is an arrangement of a product's functional elements into a number of physical building blocks (Voss and Hsuan, 2009). These building blocks are referred to as modules and can be combined and matched under certain constraints described by the architecture (Schilling, 2000). In order to reap the benefits of a modularized product program, the interfaces between the modules must be kept static. If interfaces are changed, not all modules will be combinable with one another, ultimately limiting the customizability of the program (Meyer and Lehnerd, 1997; Mortensen *et al.*, 2016). When deciding on modular decomposition, various drivers can act as the force that leads to decision-making. For products, these module drivers can be carry over, common unit, upgrading, or technical specifications (Ericsson and Erixon, 1999). When designing modularized product architectures, it can be a good idea to evaluate the value-addition of the module variants included in the final program. The value of a variant can be evaluated in terms of a trade-off between commonality with other parts and the variance delivered to the market. Non-value adding variance is, then, variance occurring in the product program that does not deliver value to the customer. A variety of indices that quantify the commonality of a product family have been proposed for use in evaluating the configuration of future families (Thevenot and Simpson, 2006).

Service architectures and modularization can help transform services from *ad hoc* activities to repeatable and configurable service offerings, but research into the topic is still new (Johnson *et al.*, 2021). Because the subject of service architectures is still new, definitions are still being formulated. However, the literature review by de Mattos *et al.* (2021) combined definitions from the service architecture literature to define a "service architecture" as a description of boundaries of the service system and a decomposition of the modules, interfaces, boundaries, and resources that define the architecture. Likewise, the definitions of the terms "modules" and "interfaces" are still largely studied on a case-by-case basis (de Blok *et al.*, 2014; de Mattos *et al.*, 2021; Eissens-van der Laan *et al.*, 2016). The combined definition posed by de Mattos *et al.* (2021) defines a service module as a set of elements that can offer value to the client and a service interface as connections among these service elements in the form of the people, information, and rules that govern information flows. This definition reflects the multidimensionality of services because service, process, physical, and human aspects have an effect on the decomposition of the service system in modules and interfaces (Eissens-van der Laan *et al.*, 2016). The effects of service modularity include the reduction of complexity, flexibility, reuse, the reduction of process time, and more (de Mattos *et al.*, 2021). Eissens-van der Laan *et al.* (2016) emphasize the importance of ensuring minimum dependencies across all dimensions in the defined service modules to improve the configurability of the modules.

The introduction of product and service architecture approaches in maintenance was proposed by Sigsgaard *et al.* (2021a). Inspired by the three domains of market, product, and production, which, when aligned, allow the greatest benefits to be reaped (Andreasen *et al.*, 1996; Mortensen *et al.*, 2010, 2011, 2016), the study introduced the three dimensions of maintenance: physical, action, and process. The architecture was then visualized through a combined overview of the three dimensions. The physical overview included a segmentation of the assets in matrix format. The action dimensions are mapped by the effects on the equipment condition from no effect to better-than-perfect maintenance, and the impact on the system is mapped from no impact to production loss. The process dimension is a mapping of the maintenance processes as they happen, showing dependencies across the life-cycle of the maintenance job (Sigsgaard *et al.*, 2021a).

MMIA – conceptual model

Introduction

This section presents the conceptual MMIA model. The main purpose of the model is to enable the reduction of non-value-adding variance and present a TO-BE architecture of modular maintenance instructions. This is achieved by creating an AS-IS overview of the existing maintenance instructions and the equipment on which they are performed. The foundation of the model is an overview of the variation of the maintenance instructions and the relationship between the two dimensions — action and physical. Because the instructions being analyzed are all assumed to be at the same stage of the process dimension (Sigsgaard *et al.*, 2021a), this dimension not included in defining the value of the variation present in the instructions.

First, the current maintenance instructions are analyzed by creating an overview of the AS-IS architecture. The understanding of an architecture in this study is based on the service architecture definition of de Mattos *et al.* (2021), wherein an architecture is the description of the boundaries of the service system and a decomposition of the modules, interfaces, and boundaries. The AS-IS overview consists of a mapping of the AS-IS maintenance instructions and the equipment on which they are used in a matrix format. The rows of the matrix represent the tasks that make up the instructions, and the columns represent groups of equipment. Based on the overview gained from the AS-IS architecture, the non-value-adding tasks can be identified, and the TO-BE architecture can be formulated. The TO-BE architecture is identified by removing the non-value-adding variance and defining a set of task modules that cover the maintenance requirements. A similar matrix format is then used to visualize and communicate the TO-BE architecture.

The details regarding the creation of the MMIA model are introduced in the following sections. These sections introduce how to achieve the elements of the architectural overview and, finally, a full overview of the model. The first section introduces how to map out the instructions. The next section introduces how to segment the equipment, and the third section introduces how to link the two dimensions together. The final section introduces a full overview of the model. The model is shown using maintenance instructions for a selection of fictional bikes.

Linking instructions

The first step in the AS-IS analysis is to link the instructions in a structured format that facilitates the analysis of the maintenance instructions, down to the individual tasks. It is difficult to compare the many unique maintenance instructions compiled from multiple tasks, making decomposition into tasks an important step. Similar to a service module, in which the module is an element or set of elements that deliver value (de Mattos *et al.*, 2021), a single task is an element or set of elements that delivers value to the maintenance goal. As such, a task or maintenance action module is a maintenance action or set of actions that can be performed independently of the remainder and is not dependent on the prior or subsequent action in the instructions. Figure 1 shows the steps in delimiting the instructions as follows: (1) the full instructions, (2) the instructions split into tasks, and (3) the similar tasks grouped. By decomposing the instructions into tasks, a full list of maintenance actions can be compiled. This is the full list of action module variants in use in the as-is architecture.

Once the instructions are separated into individual action modules, it is possible to identify similar tasks because the similarities are now clearer. The tasks are segmented into groups based on the effect of the task. At this stage, it is important to note differences that may be drivers of necessary variance (e.g. a tire change on a race bike requires a different set of tools than the same task on a mountain bike). The toolset drives the necessary variance, even though the "tire-change" operation may seem identical. It is important to ensure that these types of variations are still visible after delimitation.

Grouping equipment

The next step is to gain an understanding of the physical dimension of the maintenance. The tasks that must be performed are heavily dependent on the physical aspects of the object being maintained (Sigsgaard *et al.*, 2021a, b). In the MMIA, the physical systems are separated into groups by physical characteristics (Figure 2). For instance, bikes with hydraulic brakes are put in one group, and bikes with wire-brakes are put in another because hydraulics and wires are maintained using two significantly different methods. The grouping gives insight into how the physical parameters drive the maintenance variations identified from the tasks in the instructions. The relevance of different parameters is highly dependent on the equipment at hand. To ensure the grouping characteristics are well-defined, they are identified in collaboration with experts. The number of details obtained about the equipment should be scoped according to expert insight into what and how much maintenance variance is driven by the different physical parameters. The grouping of the equipment enables the identification of unnecessary variance across equipment that has the same parts. It is also an analysis of the maintenance requirements, providing insight into the effects of the maintenance currently being performed. Three groups were formulated for the bike example (Figure 2). The groups were formed based on gear type, the presence of suspension, required tire pressure, and types of brakes.

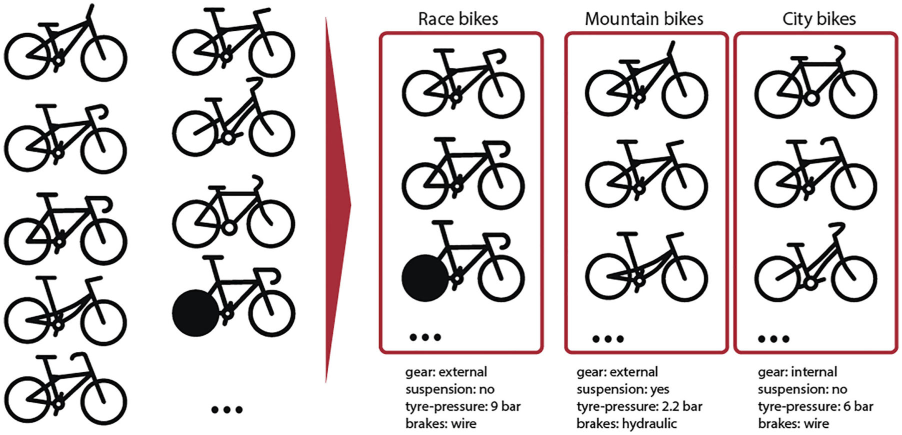
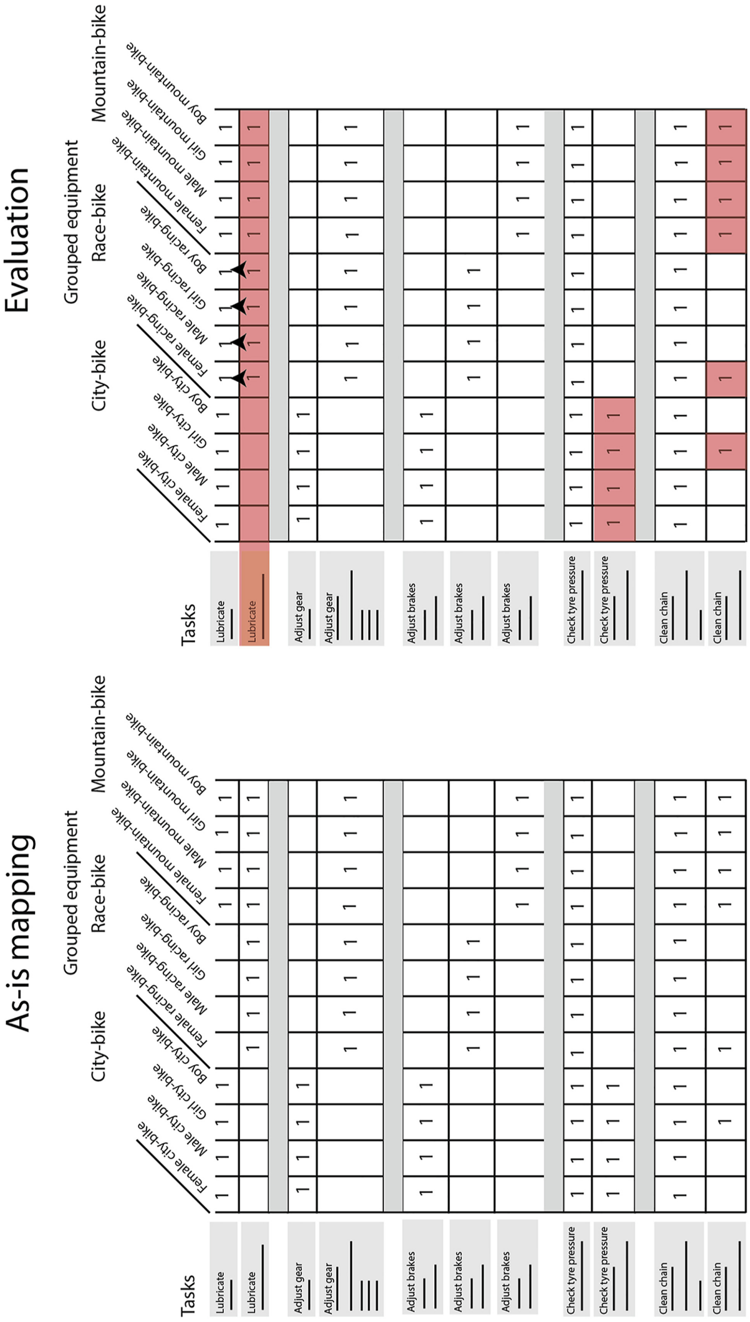


Figure 1.
Delimiting instructions into tasks and grouping. 1: The starting format of the instructions. 2: The instructions delimited into independent tasks. 3: The tasks grouped by similarities in effect



Evaluation

Tasks	Female city-bike	Male city-bike	Girl city-bike	Boy city-bike	Female racing-bike	Male racing-bike	Girl racing-bike	Boy racing-bike	Female mountain-bike	Male mountain-bike	Girl mountain-bike	Boy mountain-bike
	City-bike				Race-bike				Mountain-bike			
Lubricate	1	1	1	1	1	1	1	1	1	1	1	1
Lubricate					1	1	1	1	1	1	1	1
Adjust gear	1	1	1	1								
Adjust gear					1	1	1	1	1	1	1	1
Adjust brakes	1	1	1	1								
Adjust brakes					1	1	1	1				
Adjust brakes									1	1	1	1
Check tyre pressure	1	1	1	1	1	1	1	1	1	1	1	1
Check tyre pressure	1	1	1	1								
Clean chain	1	1	1	1	1	1	1	1	1	1	1	1
Clean chain			1		1				1	1	1	1

Grouped equipment

As-is mapping

Tasks	Female city-bike	Male city-bike	Girl city-bike	Boy city-bike	Female racing-bike	Male racing-bike	Girl racing-bike	Boy racing-bike	Female mountain-bike	Male mountain-bike	Girl mountain-bike	Boy mountain-bike
	City-bike				Race-bike				Mountain-bike			
Lubricate	1	1	1	1					1	1	1	1
Lubricate					1	1	1	1	1	1	1	1
Adjust gear	1	1	1	1								
Adjust gear					1	1	1	1	1	1	1	1
Adjust brakes	1	1	1	1								
Adjust brakes					1	1	1	1				
Adjust brakes									1	1	1	1
Check tyre pressure	1	1	1	1	1	1	1	1	1	1	1	1
Check tyre pressure	1	1	1	1								
Clean chain	1	1	1	1	1	1	1	1	1	1	1	1
Clean chain			1		1				1	1	1	1

Grouped equipment

Figure 2.
The bikes are grouped according to the characteristics of the bike parts. This allows the comparison of different types of maintenance within groups that can be maintained in the same way

The dimensional link

The link between the action dimension and the physical dimension is key to the usability of the proposed model. The main drivers of value-adding variance are the physical characteristics of the maintained equipment because different types of equipment ultimately need different maintenance. Furthermore, it is impossible to find patterns in the maintenance across different pieces of equipment without using the action dimension (Sigsgaard *et al.*, 2021b). It is rarely interesting to consider single pieces of equipment individually, because there tends to be greater savings potential when considering multiple pieces of equipment. For example, one might want to consider how the differences between mountain bike gear systems and city bike gear systems affect the maintenance of the bikes and how this can be optimized. The grouping of equipment is based on the possibility of executing operational tasks on that group. For example, if bikes are grouped according to suspension type, "check for suspension leakage" is expected to apply to all bikes within that group. All equipment in the group should have similar maintenance requirements. It is impossible to achieve a group that is similar in every aspect. As such, the limitations of the grouping should be made clear when the TO-BE situation is formulated.

The relationship between the dimensions is the key element of the model because it enables different patterns to be observed. The relationships reveal the use of tasks across the equipment and equipment groups. If a task has a high degree of re-use across equipment types, it is likely very generic and could form the basis of new modules. An example can be seen in Figure 3: "Clean chain 1" is used on most bikes, while 2, 3, and 4 are only used on one or two bikes. Diagonality in the matrix implies high variance, while horizontality implies a high degree of reuse. Thus, the patterns in the matrix can be used as indicators of what should be investigated further.

Architecture

The MMIA consists of two architectural overviews: the AS-IS architecture and the TO-BE architecture. Both are based on the same matrix format, with tasks as rows and grouped equipment as columns. The input tasks and equipment groups are the results described in the previous three sections. The AS-IS architecture view shows the action module variants

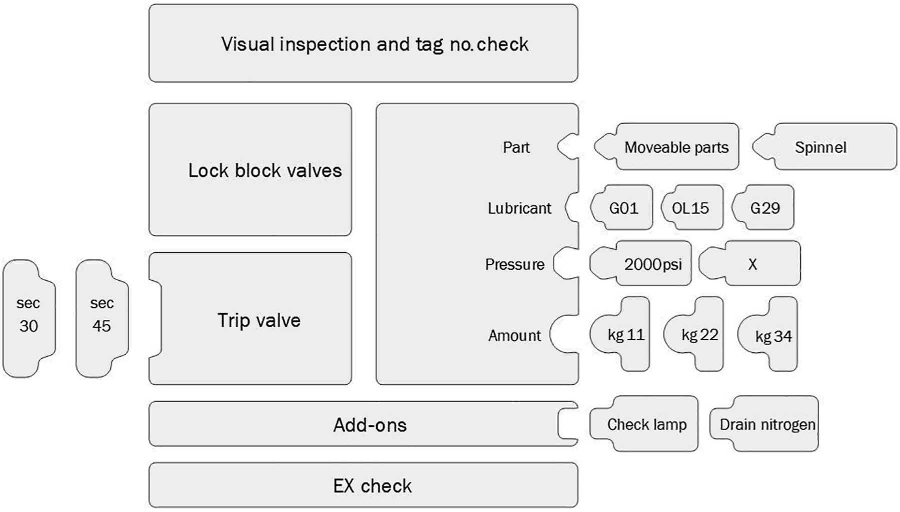


Figure 3.
An example of a pattern for chain-cleaning tasks

currently in use against the physical dimensions. The TO-BE architecture is thus a representation of the reconfigured architecture in which all action module variants are value-adding to the physical dimension.

The AS-IS architecture for the example of bicycle maintenance is shown in Figure 4. The matrix on the left side of the figure shows the maintenance performed on the bicycles. A "1" in the matrix indicates that the maintenance task is used for the given equipment group. The groups along the columns were created based on the commonality of the structure and parts of the bikes. This overview makes it possible to identify variations and similarities in the instructions for each bike variant because bikes consisting of the same parts can be maintained in the same way.

Using the AS-IS overview, the value addition of the variants can be evaluated. Value addition is evaluated by comparing the variance found in the proposed actions to the value delivered to the physical dimension. If there is no significant value in the variance in the output of the maintenance task, the task variance is marked as non-value-adding. The right-hand side of Figure 4 shows examples of non-value-adding variants that can be eliminated. At the top of the matrix are two variants of lubrication tasks applied throughout all three product groups. Because the maintenance within a group can be the same, the extra variant adds complexity without adding value. Therefore, one of the task variants can be chosen, and the other eliminated. A similar situation can be seen for "check tire pressure" and "clean chain." In both cases, a variant can be removed because one task is enough. For the "adjust gear" tasks, two variants of the task are in use. In this case, the use within the equipment groups is more inconsistent. To enhance the performance of the maintenance, both variants are needed, so both are kept.

When the non-value-adding variance is identified, it can be removed to create an improved TO-BE situation (Figure 5). The TO-BE architecture is then used as the basis for creating new maintenance instructions by combining the task modules. Combinations of the modules can make every instruction unique while still consisting of well-defined modules that are easily updated whenever the requirements change. A requirement change could come from new parts being introduced or from a change in the product. When the instruction requires an update, only the affected module must be updated rather than the entire instruction.

Case study – application of the model

To test the applicability of the model, a case study was conducted at a large production company. The company operates large, offshore, continuous production plants, which require an increasing amount of maintenance due to the age of the plants and the rough offshore environment. The plants consist of a large number safety-critical valves, and their functionality is essential to running safe plants. These valves are maintained to ensure their functionality, but the instructions for valve maintenance have a large amount of variation. This has made them difficult to manage because a full overview of the actions to be taken regarding the safety-critical valves is not available. The scope of the case study includes a total of 5,636 maintenance instructions for 1,941 valves across four assets.

The first step in the study was to collect the data. Three data types were the cornerstones of the analysis: the physical characteristics of the equipment, the historical data on the execution of maintenance, and the maintenance instructions. The maintenance instructions were 1–10 pages long and consisted mainly of tasks in a bullet-point format. A few instructions were so different from the remainder that they were marked as outliers and excluded from the scope. This decision was made in collaboration with experts on company maintenance, and the jobs were rare jobs that only occurred with large intervals and involved many steps outside of the normal process.

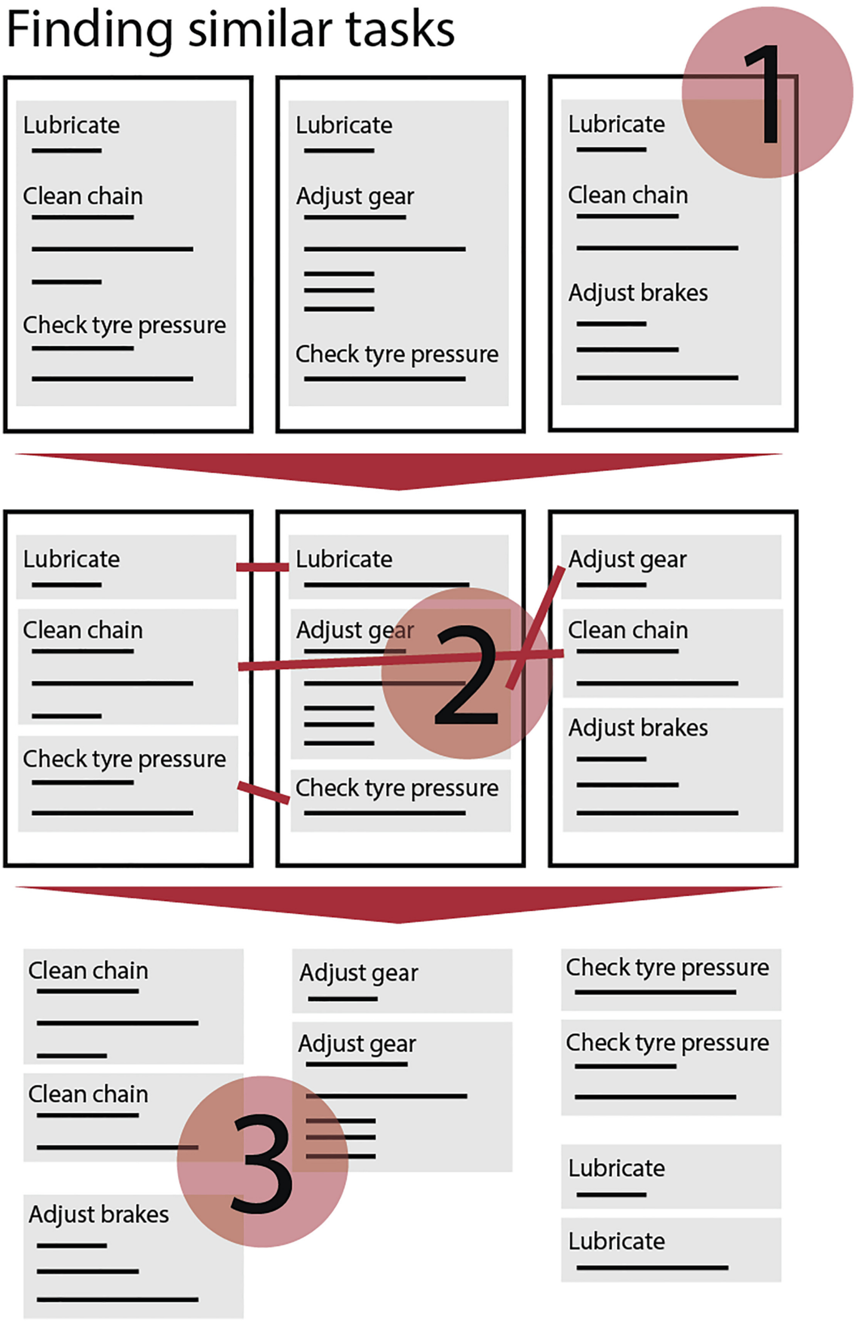
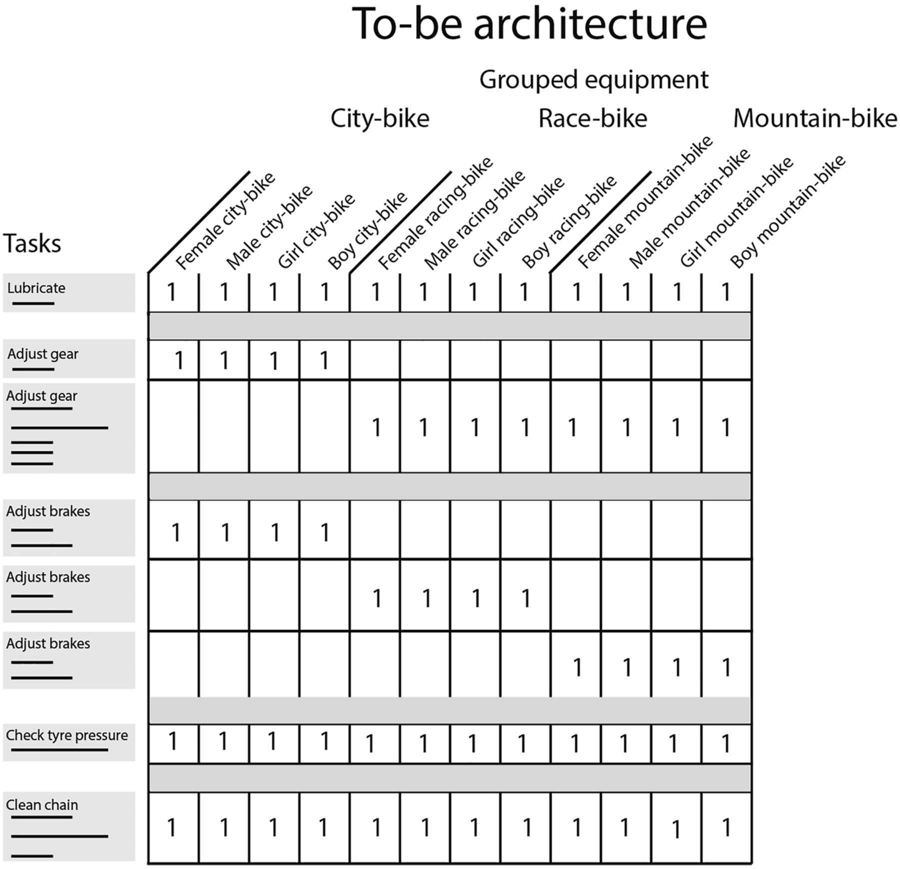


Figure 4.
The AS-IS architecture for the example of bike maintenance. LEFT: The complete AS-IS overview. RIGHT: Identifying non-value-adding variance in the AS-IS architecture



Tasks	Female city-bike	Male city-bike	Girl city-bike	Boy city-bike	Female racing-bike	Male racing-bike	Girl racing-bike	Boy racing-bike	Female mountain-bike	Male mountain-bike	Girl mountain-bike	Boy mountain-bike
Lubricate	1	1	1	1	1	1	1	1	1	1	1	1
Adjust gear	1	1	1	1								
Adjust gear					1	1	1	1	1	1	1	1
Adjust brakes	1	1	1	1								
Adjust brakes					1	1	1	1				
Adjust brakes									1	1	1	1
Check tyre pressure	1	1	1	1	1	1	1	1	1	1	1	1
Clean chain	1	1	1	1	1	1	1	1	1	1	1	1

Figure 5.
The TO-BE architecture. The non-value-adding variants have been removed

Due to the large amount of text, the process was automated. A database of all the tasks was created by running a script in Python that separated the bullet points into rows by identifying the bullet point character and line changes and making a separate entry between the line changes and bullets. All tasks were then compared to identify similarities. If two or more tasks were the same in every respect, only one version was included in the final overview. Due to the large amount of data at this stage, only tasks that were completely similar in every respect were marked as similar. Somewhat similar tasks were kept separate. The tasks were then imported to a spreadsheet. This final spreadsheet included 413 unique tasks.

To understand whether the content of the tasks was unique or consisted of variations of the same tasks, the entire library of tasks was manually categorized. This was done by reading each task and assigning it to the appropriate group. One group could, for instance, be "visual inspection" — each time a task had "visual inspection" in the text or something related to visual inspection, it would be included in the corresponding category. Figure 6 shows an example of the variants of visual inspection tasks that were covered by one module in the to-be architecture. The categorization and grouping of the tasks was a type of subjective analysis that would not necessarily produce the same output if repeated by someone else. Some tasks were more challenging to group, either because they were unique to

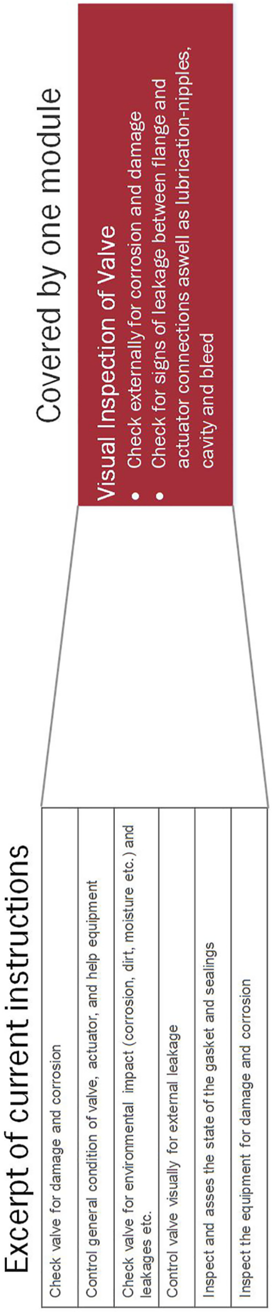


Figure 6.
Examples of the variance in as-is instructions that could have been covered with a single module

a certain kind of valve or because the required knowledge was not available. To ensure that the assumptions made were correct, the identified opportunities for variance minimization were verified by maintenance experts from the case company, as shown in Figure 4, right.

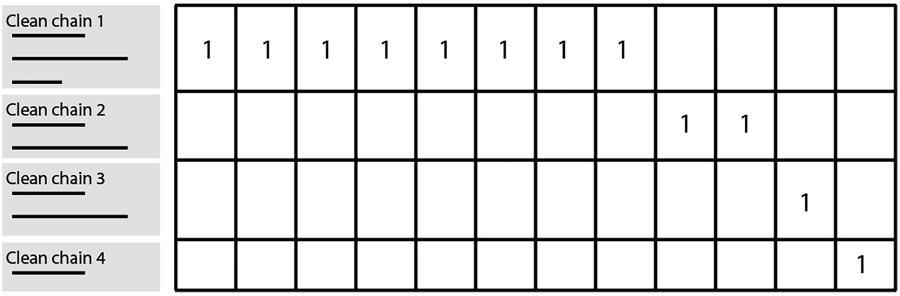
Historical data were used to form the link between the action dimension data and the physical dimension. The historical data were obtained from the case company's CMMS. The system contains reports on the findings, states, and repairs performed during maintenance on each valve; information on when each valve was maintained and what set of instructions was used; and the physical characteristics of the valves, such as type, size, and medium. In building a data model from the historical data, the link between the executed maintenance tasks and the actual valves and their characteristics was formed.

The AS-IS architecture was presented in a workshop with company experts with two objectives in mind: (1) verify the collected data, the cleaning processes, and the manual categorization process and assumptions and (2) develop the TO-BE architecture by collaboratively eliminating non-value-adding variation.

The AS-IS architecture overview revealed that none of the physical parameters drove the variation found in the instructions. This showed that it was possible to use the same task for multiple valves. Many of the tasks were variations for situations involving exactly the same physical characteristics. For instance, when a worker checks the visibility of the tag number of a valve, a new method is not needed for every valve. Inputs from the experts indicated that many of the tasks were outdated and included unwanted procedures and content. The categorization of the tasks enabled the experts to quickly point out entire categories and define when the tasks needed to be the same or unique. In collaboration with the experts, the number of tasks that could be modularized was 224, leading to 20 modules and covering 60% of the instructions. This covered the maintenance instructions for 83% of valves. The modules covered the same tasks as before but were simplified and improved in quality. The collaboration with the company maintenance experts ensured that the resulting task modules were all feasible. The remaining 40% of tasks (189) could not directly be modularized, and it was necessary to keep them AS-IS. The final modules and examples of the module variants can be seen in Figure 7. The figure shows the modules in the center and the module variants as puzzle blocks whose interfaces will only fit in specific modules.

Discussion

The study described in this paper shows the potential of simplifying maintenance instructions by reducing the non-value-adding variants and modularizing the remaining tasks to achieve maintenance configurability. The modular instructions reduced complexity by reducing non-value-adding variance, while the maintenance can still be differentiated to suit variations in the physical requirements of the equipment.



Clean chain 1	1	1	1	1	1	1	1	1				
Clean chain 2									1	1		
Clean chain 3											1	
Clean chain 4												1

Figure 7. Visual representation of the modules (central blocks) and examples of the module variants (smaller blocks) defined for the modular maintenance instruction architecture at the case company

The proposed method, the MMIA, is a systematic method applied with the goal of reducing the non-value-adding variety in maintenance instructions so as to achieve a greater overview of maintenance actions. The advantage of using the proposed MMIA is that it provides insights into the current situation, which provides an understanding of the changes necessary to move from an overly complex AS-IS to the desired, optimal TO-BE architecture. The method is based on product and service architecture approaches, as well as the definitions of maintenance and service architecture and modularization (de Mattos *et al.*, 2021). Similar to service architectures, the maintenance architecture is based in multiple dimensions, being dependent on the physical, action, and process dimensions. However, the client, in maintenance, is instead the operator of the facilities, who is also responsible for the maintenance. The application of the approaches and definitions of architectures and modules indicates the usefulness of these in maintenance management, but clear definitions of maintenance architectures and modularization are still in the early stages. This also means that the definitions of commonality indices (Thevenot and Simpson, 2006) and module drivers (Ericsson and Erixon, 1999) are not yet fully formed for maintenance modularization. This is reflected in the needs of the results from the method to be evaluated by experts in the case company. More studies beyond the case-based scope are needed to define these aspects in larger scopes and in other companies and industries. A clear drawback of the proposed MMIA is that it is necessary to include a large number of experts in order to verify the findings. This is an extra step that requires a great deal of time and resources because the experts must do much work as well. However, these experts were already spending large amounts of time reading and understanding the various maintenance instructions. Improving the maintenance instruction architecture will improve the time spent on analyzing the maintenance being performed, providing a return on the time invested.

The results of the case study showed that it was possible to gain repeatability and reduce the complexity of the maintenance instructions by minimizing the non-value-adding variation in the maintenance. The resulting modules and module variants were defined by the value the actions delivered in terms of the physical characteristics of the production equipment. Because the decomposition of the tasks was performed to minimize dependencies across the modules (Eissens-van der Laan *et al.*, 2016), the interfaces between the modules are more defined by the requirements of the physical dimensions, i.e. equipment characteristics, than by the dependencies in the action dimension. This study focused more heavily on the development and configuration of the TO-BE architecture than the longer-term use and upkeep of the architecture. More longitudinal studies are needed to show whether the longer-term benefits of the TO-BE architecture reflect those outlined in service architecture studies, such as the reduction of process time, quality improvement, and possibilities for work improvement (de Mattos *et al.*, 2021), or product architecture studies, such as the reduction of development time, a quicker response to changes, and the reduction of costs (Harlou, 2006).

The method differs from those identified from the maintenance instruction research because it focuses on reducing non-value-adding variety. As such, it can be used as an input or starting point for the methods highlighted in the literature section, such as the TOAMS (Huang *et al.*, 2015), customized maintenance documents (Huang *et al.*, 2014), or AR implementation (Fiorentino *et al.*, 2014; Havard *et al.*, 2021; Mourtzis *et al.*, 2020). From a maintenance architecture perspective, the method especially focuses on the links between the physical and action dimensions (Sigsgaard *et al.*, 2021a). The process dimension was excluded from the MMIA. Because the instructions are all considered to be in the planning stage, they are all considered to have the same level of maturity. The modularization of the instructions can, similarly, be an input into the action view, as proposed by Sigsgaard *et al.* (2021a). Future longitudinal studies should also focus on the effects of the process dimension when the architecture is used to plan, perform, and evaluate the maintenance.

The study of the proposed MMIA method indicates the usefulness of the method, but it is based only on one case company with one type of equipment. Further work should therefore focus on evaluating the model by testing it in other contexts, on other types of equipment, and on a larger scale. Such work would also be a further contribution to the applicability and definition of maintenance architectures and modularization initially proposed by Sigsgaard *et al.* (2021a).

Conclusion

This paper presents a method for evaluating the variations in maintenance instructions by linking the instructions to the dimensions of maintenance. The connection across the dimensions made it possible to identify and remove non-value-adding variation in the maintenance tasks. A case study was used to show how a Modular Maintenance Instructions Architecture (MMIA) can be achieved using the proposed method. The MMIA breaks down maintenance instructions into independent, comparable tasks and links this to equipment to enable an evaluation of the value addition of the task variation. The case study resulted in a reduction of 224 tasks to 20 unique tasks. The 20 unique tasks were able to cover the maintenance of 83% of the pieces of equipment studied in the scope.

References

Agergaard, J.K., Sigsgaard, K.V., Mortensen, N.H., Ge, J., Hansen, K.B. and Khalid, W. (2021), "Standardising maintenance jobs to improve grouping decision making", *Proceedings of the Design Society*, Vol. 1, pp. 2701-2710, doi: 10.1017/PDS.2021.531.

Andreasen, M.M., Hansen, C.T. and Mortensen, N.H. (1996), "The structuring of products and product programmes", *2nd WDK Workshop on Product Structuring*, Design Society, pp. 15-43.

Blessing, L.T.M. and Chakrabarti, A. (2009), *DRM, a Design Research Methodology, DRM, a Design Research Methodology*, Springer, London, doi: 10.1007/978-1-84882-587-1.

Dansk Standard (2016), DS/EN ISO 14224: petroleum, petrochemical and natural gas industries-Collection and exchange of reliability and maintenance data for equipment.

de Blok, C., Meijboom, B., Luijkx, K., Schols, J. and Schroeder, R. (2014), "Interfaces in service modularity: a typology developed in modular health care provision", *Journal of Operations Management*, Vol. 32 No. 4, pp. 175-189, doi: 10.1016/J.JOM.2014.03.001.

de Mattos, C.S., Fettermann, D.C. and Cauchick-Miguel, P.A. (2021), "Service modularity: literature overview of concepts, effects, enablers, and methods", *Service Industries Journal*, Vol. 41 Nos 15-16, pp. 1007-1028, doi: 10.1080/02642069.2019.1572117.

Eissens-van der Laan, M., Broekhuis, M., van Offenbeek, M. and Ahaus, K. (2016), "Service decomposition: a conceptual analysis of modularizing services", *International Journal of Operations and Production Management*, Vol. 36 No. 3, pp. 308-331, doi: 10.1108/IJOPM-06-2015-0370/FULL/HTML.

Ericsson, A. and Erixon, G. (1999), "Controlling design variants: modular product platforms", pp. 145, available at: http://www.google.is/books?hl=en&lr=&id=M-SPpezS7WkC&pgis=1

Fiorentino, M., Uva, A.E., Gattullo, M., Debernardis, S. and Monno, G. (2014), "Augmented reality on large screen for interactive maintenance instructions", *Computers in Industry*, Vol. 65 No. 2, pp. 270-278, doi: 10.1016/J.COMPIND.2013.11.004.

Harlou, U. (2006), *Developing Product Families Based on Architectures, Contribution to a Theory of Product Families*, Technical University of Denmark.

Harris, P.J. (1994), "An expert systems technology approach to maintenance proficiency", *Robotics and Computer Integrated Manufacturing*, Vol. 11 No. 3, pp. 195-199, doi: 10.1016/0736-5845(94)90034-5.

Havard, V., Baudry, D., Jeanne, B., Louis, A. and Savatier, X. (2021), “A use case study comparing augmented reality (AR) and electronic document-based maintenance instructions considering tasks complexity and operator competency level”, *Virtual Reality*, Springer Science and Business Media Deutschland GmbH, Vol. 25 No. 4, pp. 999-1014, doi: 10.1007/S10055-020-00493-Z.

Horn, R.E. (1993), “Structured writing at twenty-five”, *Performance + Instruction*, Vol. 32 No. 2, pp. 11-17, doi: 10.1002/pfi.4170320206.

Huang, Y., Gardoni, M., Wang, X. and Amadou, C. (2014), “A knowledge integration methodology for developing customized maintenance documents”, *Interdisciplinary Journal of Information, Knowledge, and Management*, Vol. 9, pp. 175-191, doi: 10.28945/2080.

Huang, Y., Wang, X., Gardoni, M. and Coulibaly, A. (2015), “Task-oriented adaptive maintenance support system”, *Journal of Computing and Information Science in Engineering*, Vol. 15 No. 3, JCISE-14-1106, doi: 10.1115/1.4030077.

Jacobs, R.L. (2017), “Knowledge work and human resource development”, *Human Resource Development Review*, Vol. 16 No. 2, pp. 176-202, doi: 10.1177/1534484317704293.

Johnson, M., Roehrich, J.K., Chakkol, M. and Davies, A. (2021), “Reconciling and reconceptualising servitization research: drawing on modularity, platforms, ecosystems, risk and governance to develop mid-range theory”, *International Journal of Operations and Production Management*, Vol. 41 No. 5, pp. 465-493, doi: 10.1108/IJOPM-08-2020-0536.

Løkkegaard, M., Mortensen, N.H. and McAloone, T.C. (2016), “Towards a framework for modular service design synthesis”, *Research in Engineering Design*, Vol. 27 No. 3, pp. 237-249, doi: 10.1007/S00163-016-0215-6.

Meyer, M.H. and Lehnerd, A.P. (1997), *The Power of Product Platforms*, The Free Press, Washington, DC.

Meyer, M.H. and Utterback, J.M. (1992), “The product family and the dynamics of core capability”, *Sloan Management Review*, Vol. 34 No. 3, pp. 29-47.

Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P., Lindschou, C. and Frobenius, S. (2010), “Making product customization profitable”, *International Journal of Industrial Engineering: Theory Applications and Practice*, Vol. 17 No. 1, pp. 25-35.

Mortensen, N.H., Hansen, C.L., Hvam, L. and Andreasen, M.M. (2011), “Proactive modeling of market, product and production architectures”, *ICED 11-18th International Conference on Engineering Design - Impacting Society Through Engineering Design*, Vol. 4, pp. 133-144.

Mortensen, N.H., Hansen, C.L., Løkkegaard, M. and Hvam, L. (2016), “Assessing the cost saving potential of shared product architectures”, *Concurrent Engineering Research and Applications*, Vol. 24 No. 2, pp. 153-163, doi: 10.1177/1063293X15624133.

Mortensen, N.H., Bertram, C.A. and Lundgaard, R. (2019), “Achieving long-term modularization benefits: a small- and medium-sized enterprise study”, *Concurrent Engineering Research and Applications*, Vol. 27 No. 1, pp. 14-27, doi: 10.1177/1063293X18803145.

Mourtzis, D., Angelopoulos, J. and Panopoulos, N. (2020), “A framework for automatic generation of augmented reality maintenance & repair instructions based on convolutional Neural networks”, *Procedia CIRP*, Vol. 93, pp. 977-982, doi: 10.1016/J.PROCIR.2020.04.130.

Otto, K., Hölttä-Otto, K., Simpson, T.W., Krause, D., Ripperda, S. and Moon, S.K. (2016), “Global views on modular design research: linking alternative methods to support modular product family concept development”, *Journal of Mechanical Design, Transactions of the ASME*, Vol. 138 No. 7, MD-15-1628, doi: 10.1115/1.4033654.

Pham, D.T., Dimov, S.S. and Peat, B.J. (2000), “Intelligent product manuals”, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 214 No. 5, pp. 411-419, doi: 10.1243/0954405001518198.

Pitblado (1991), *Plant Engineer’s Reference Book*, Butterworth-Heinemann, Oxford.

Schilling, M.A. (2000), “Toward a general modular systems theory and its application to interfirm product modularity”, *The Academy of Management Review*, Vol. 25 No. 2, p. 312, doi: 10.2307/259016.

Setchi, R.M., Pham, D.T. and Dimov, S.S. (2006), "A methodology for developing intelligent product manuals", *Engineering Applications of Artificial Intelligence*, Vol. 19 No. 6, pp. 657-669, doi: 10.1016/J.ENGAPPAI.2006.06.004.

Sigsgaard, K.V., Soleymani, I., Mortensen, N.H., Khalid, W. and Hansen, K.B. (2021a), "Toward a framework for a maintenance architecture", *Journal of Quality in Maintenance Engineering*, Emerald Group Holdings, Vol. 28 No. 2, pp. 474-490, doi: 10.1108/JQME-01-2020-0004/FULL/PDF.

Sigsgaard, K.v., Agergaard, J.K., Mortensen, N.H. and Soleymani, I. (2021b), "Data-driven systematic evaluation of preventive maintenance performance", *Proceedings - Annual Reliability and Maintainability Symposium*, 2021-May, doi: 10.1109/RAMS48097.2021.9605706.

Simpson, T.W., Jiao, J.R., Siddique, Z. and Hölttä-Otto, K. (2014), *Advances in Product Family and Product Platform Design: Methods & Applications*, Springer, New York, doi: 10.1007/978-1-4614-7937-6.

Thevenot, H. and Simpson, T. (2006), "Commonality indices for product family design: a detailed comparison", *Journal of Engineering Design*, Vol. 17 No. 2, pp. 99-119, doi: 10.1080/09544820500275693.

Toscano, L. (2000), "Electronic technical manual development", *IEEE Aerospace and Electronic Systems Magazine*, Vol. 15 No. 8, pp. 22-24, doi: 10.1109/62.861769.

Ulrich, K. (1995), "The role of product architecture in the manufacturing firm", *Research Policy*, Vol. 24 No. 3, pp. 419-440, doi: 10.1016/0048-7333(94)00775-3.

Voss, C.A. and Hsuan, J. (2009), "Service architecture and modularity", *Decision Sciences*, Vol. 40 No. 3, pp. 541-569, doi: 10.1111/J.1540-5915.2009.00241.X.

Wilson, S.A. and Perumal, A. (2009), *Waging War Complexity on Complexity Costs*, McGraw-Hill Education Europe.

Corresponding author
Kristoffer Vandrup Sigsgaard can be contacted at: krvsig@mek.dtu.dk