



---

# Concept for Rule-Based Information Aggregation in Modular Production Plants

Christoph Geng, Natalia Moriz, Andreas Bunte and Henning Trsek

---

## Abstract

In the context of Industrie 4.0, (Self-)adapting Cyber-Physical Production Systems (CPPS) offer a solution for production facilities to adapt to changing market requirements. The operation of a CPPS requires information of the entire plant at any time, which can usually be derived by aggregating the individual information of the production modules of a CPPS (information models). Even if the information models are designed following standardized guidelines, the aggregation of these individual information models often needs to be done manually, which requires a lot of effort and is error prone. To achieve an automated information aggregation, this paper presents a novel concept for modular production plants. The main aspect of the concept is the use of a *Rule Engine* for the processing of information models in order to create aggregated plant information. This *Rule Engine* uses a classification method for plant components in order to enable standardization. The challenge of this work is to design a *Rule Engine* that is suitable for different aspects of CPPS.

---

## Keywords

Information Modeling · Information Aggregation · Modular Production Systems · Cyber Physical Production Systems · Rule-based Information Aggregation

---

C. Geng (✉) · N. Moriz · A. Bunte · H. Trsek  
inIT – Institute Industrial IT, Ostwestfalen-Lippe University of Applied Sciences and Arts, Lemgo,  
Germany  
e-mail: [christoph.geng@th-owl.de](mailto:christoph.geng@th-owl.de); [natalia.moriz@th-owl.de](mailto:natalia.moriz@th-owl.de); [andreas.bunte@th-owl.de](mailto:andreas.bunte@th-owl.de);  
[henning.trsek@th-owl.de](mailto:henning.trsek@th-owl.de)

## 1 Introduction

In the context of Industrie 4.0, where a high degree of flexibility in product variants is demanded, production facilities must be able to adapt to changing conditions. Cyber Physical Production Systems (CPPS), which are characterized, amongst other things, by a modular plant design, can be used for this purpose [12]. Within CPPS, many different modules, often produced by different machine-builders, work together in a variety of combinations [16]. The operation of CPPS requires information of the entire plant. In a plant, it can be distinguished between different hierarchy levels: the plant itself, production lines, production modules and components. The information aggregation for the whole plant can usually be started by aggregating the individual information of the production modules, which are often represented by information models [2]. Then, this procedure can be repeated for the higher levels of the plant hierarchy. However, the aggregation of individual information models is challenging, because no standardized procedures or methods exist. Additionally, even within components of the same manufacturer, there are often no standardized information models. Furthermore, the integration of new information due to retro-fitting has to be considered in terms of information aggregation. Today, the process of information aggregation is mostly performed manually, as methods that deal with the aforementioned challenges are missing. The manual information aggregation has a number of disadvantages, e.g. it is error-prone and requires a huge amount of time.

As a solution approach, this paper introduces a concept for rule-based information aggregation for modular production plants by using a *Rule Engine*, which makes use of a classification method for plant components. A crucial part of the *Rule Engine* is a *Rule Set* which is based on the principle of logical inference. This *Rule Set* will be used to determine, which aggregated information can be generated. Within this paper, the concept is applied to a use case from the area of skill-based engineering with a production plant.

The remainder of the paper is organized as follows. First, the relevant state of the art is discussed in terms for existing approaches. After this, a novel concept is introduced and applied to a use case. Finally, the main observations are summarized and an outlook on future work is given.

---

## 2 State of the Art

Research in the area of information aggregation can mainly be found in the field of social sciences, economic sciences and sensor networks [3, 4, 14, 18]. In social sciences, it is shown how information about people can be aggregated in order to analyze their behavior. The economic sector uses information aggregation to draw conclusions about the behavior of stock markets. In the context of sensor network, for example, energy savings can be achieved through data aggregation. A classification for data aggregation systems is introduced in [8]. It focuses on systems that represent selected elements of the information model of distributed systems in their own address space. The classification consists of an

architecture and a capability model. OPC UA has a huge potential for the implementation of such a model. It can be used as a semantic information carrier and can be integrated into the introduced architecture without much effort. The work at hand can be embedded into the following categories of a functional classification of [8]: information centralization, central control of distributed CPS and the data and information processing. The demand of a rule set is uncontroversial in the literature. Some works suppose that the creation of the aggregation rule sets is a task of the technology [13, 17]. Other works assume it as platform independent rule sets as mentioned in the works [2, 5]. However, works with more details about the definition of such rule sets cannot be found in the literature.

In contrast to focusing on production rules, other works use different types of knowledge modeling to represent the calculation of Key Performance Indicators (KPIs) in CPPS. Nevertheless, there are some parallels to the approach presented in this paper. Such as Kumagai et al. [9], there needs to be standardized information models to ensure the correct aggregated information. Kumagai et al. do not take modular systems into account, but they define different levels of abstraction. The aim of Walzel et al. [15] is the closest to this work, since they automatically aggregate KPIs in modular CPPS. They use ontologies to represent the knowledge and engineering data as source of information instead of information models. However, they are limited to KPIs and do not use production rules. So, there are some works available, but they are focusing on limited aspects regarding CPPS. The analysis of the state of the art has shown that there is no generic method for the aggregation of information models covering different aspects in CPPS.

---

### 3 Concept for Rule-Based Information Aggregation

This section introduces a concept of how information from a modular production plant can be managed and aggregated. The goal of the concept is to make information of the production plant available in different degrees of abstraction. For this purpose, information is aggregated and combined to a more abstract form. The degree of abstraction is arbitrary. This means that there can be different levels of aggregation, ranging from the lowest level (e.g. single sensor values) to performance indicators of whole CPPS. An example of information aggregation is that the total energy consumption of the entire plant is determined and displayed from the information of the individual production modules of a plant. Another example is the creation of skills in the area of skill-based engineering.

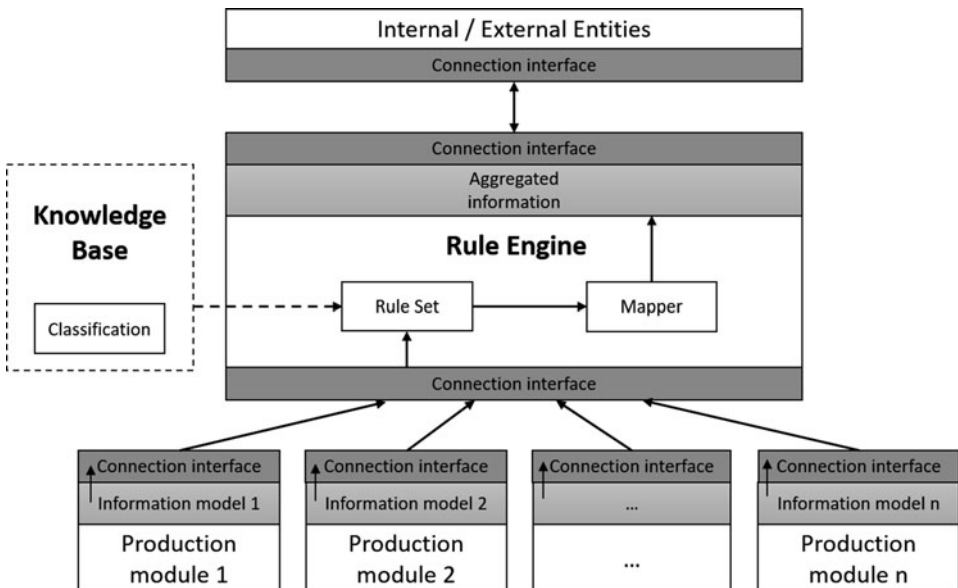
The concept can be applied independently from the technology used: No specific information modeling methods or specific programming languages have to be used. Although it is not specified which information modeling method has to be used, the concept supposes that separate information models are created using consistent methods (e.g. OPC UA [10] or AutomationML [1]). All properties (e.g. sensor values) must be mapped in these information models. The information models must be made available to further entities via suitable connection interfaces.

### 3.1 Structure of the Concept

The central part of the concept is a building block that has access to all available information models of the production plant via a connection interface. This building block is called the *Rule Engine*. This part of the concept should enable other entities (internal and external) to connect to it and thus read all available information. These information must be available in a standardized form. For this purpose, a classification of the system components is needed. One possible method is described in more detail in Sect. 3.2. The main function of the concept, however, is to aggregate plant information using the *Rule Engine*, which consists of two modules.

The first module is called *Rule Set* and consists of rules. The purpose of these rules is to determine, what type of aggregated information can be generated on the basis of existing information. To do this, the existing information is analyzed which means that information is categorized and interdependencies are identified. This determines which aggregated information can be created, based on the logical inference of rules. Details about this will be discussed in Sect. 3.3.

The second module is called the *Mapper*. This module is responsible for ensuring that the aggregated information determined via the *Rule Set* is actually generated (e.g. by storing values in a variable or creating method calls). Figure 1 shows the structure of the concept. It should be noted that the *Knowledge Base* is only involved in the development phase of the *Rule Set*.



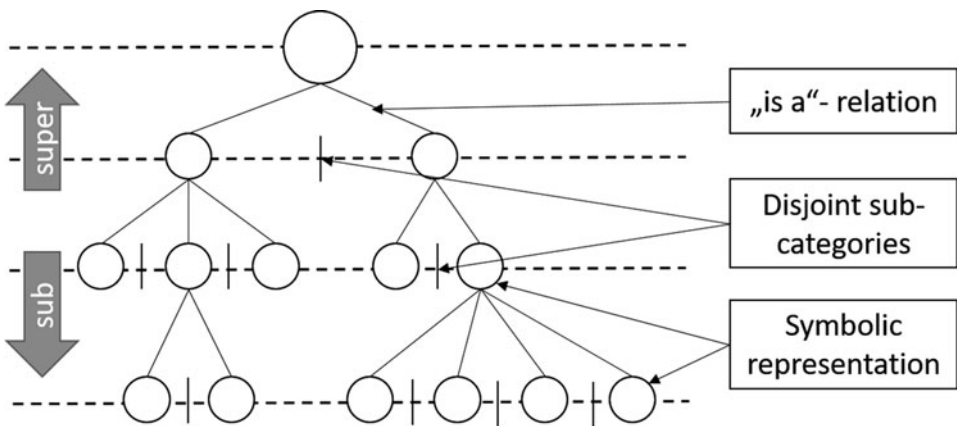
**Fig. 1** Overview of the information aggregation concept

### 3.2 Classification Method

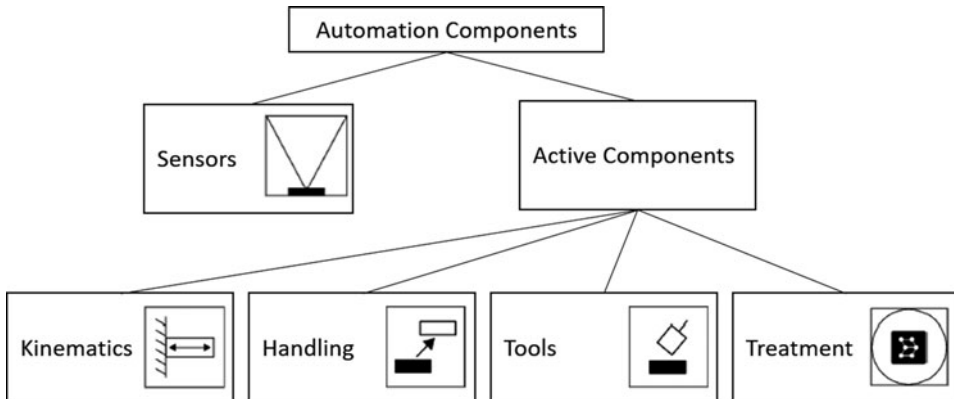
As described in Sect. 3.1, in this concept, information is aggregated using a *Rule Engine*. However, this requires that the existing information is available in a standardized and uniform representation. This can, for example, be based on norms or industry standards. In this paper, a classification method is applied, which is based on various DIN standards and VDI guidelines and is presented in [6].

For the classification of system components, a description form must be used. A description form is a kind of formalism which describes the characteristics and capabilities of the plant components. It must meet the requirements of *uniformity*, *accuracy*, *completeness* and *comprehensibility*. A taxonomy is used for this purpose. The principle here is that generic main categories are created at the top level. To specify an element, main categories are divided into several subcategories. Each subcategory is connected to the main category via a “is a”-relation and inherits its properties. Each subcategory also adds further characteristics. Subcategories must be disjoint so that elements can be assigned uniquely. This satisfies the property of *uniformity*. The subdivision into subcategories is continued until the desired level of detail is reached in order to satisfy the property of *accuracy*. The *completeness* is achieved by ensuring that all components can be classified. Making it possible to grasp the characteristics of a category as quickly as possible, ensures the requirement of *comprehensibility*. For this purpose, each category is assigned a meaningful symbol. This refers to both the naming of the categories and the visual representation [7]. Figure 2 shows the principle of a taxonomy.

In order to apply the taxonomy to plant components, a distinction is made on the first level of the hierarchy between *sensor* and *active components*. *Sensors* include those elements that are used to record the state of the system. *Active components* are those parts of the plant that contribute to the function of the system. This category in turn can be divided into the subcategories *kinematics*, *handling*, *tools* and *treatment*. These



**Fig. 2** Taxonomy of categories [6]



**Fig. 3** Categories used for a taxonomy [6]

subcategories are further specified, as it can be seen from [6], so that a detailed taxonomy is available at the end. Figure 3 shows a section with the first two levels of the categories for the taxonomy [6].

Categorizing plant components ensures that mandatory information about these components are made available. This can be, for example, information about the capabilities of a particular component (e.g. the direction of movement of a linear axis). This mandatory information is evaluated by the *Rule Engine*, which is described in the next section.

### 3.3 Rule Engine

The *Rule Engine* is the core element of the concept presented in this paper. The process of information aggregation can be divided into two parts which are performed by the *Rule Set* and the *Mapper* respectively. This section explains how the two modules of the *Rule Engine* work.

The module *Rule Set* consists of rules that are used to analyze the input information and determine, which aggregated information should be generated. The principle of deductive reasoning is applied to the rules, according to which causes and laws result in conclusions. Causes and laws are derived from information models or knowledge bases which differ depending on the viewed aspect of CPPS. A specific part of knowledge bases are classifications of plant components, which were introduced in Sect. 3.2. These classifications guarantee the existence of certain characteristics of components (mandatory information). The conclusions represent possible aggregated information. This information, in turn, can serve as causes for further rules. Thus, information about the overall state of the plant is to be obtained from the individual information models.

The *Mapper* uses the results from the *Rule Set* and implements the aggregated information. Therefore, the *Mapper* maps inferred symbols from the rules to the functions that implement the indicated methods. To enable this kind of mapping, the *Rule Set* and

the *Mapper* have to be compatible, so if the *Rule Set* changes, the *Mapper* might be adapted since all symbols of the rules have to be implemented in the *Mapper*. Because all incoming information is provided by an information model or a knowledge base, typically, the *Mapper* does not have to be adapted if it is transferred to another CPPS with the same *Rule Set*. So, it is as generic as the *Rule Set*. For example, if information of the power consumption is aggregated, the *Mapper* creates a node in the information model and performs the calculation of its values. Another example are functions of the modules that are aggregated by the *Rule Set*. Here, the *Mapper* is responsible for the method calls of the modules' function.

---

## 4 Concept Implementation for a Specific Use Case

This chapter explains how the concept described in Sect. 3 can be applied to a use case. For this purpose, a part of a production plant from the SmartFactoryOWL is considered. Since principles of skill-based engineering are demonstrated on this production plant, the application of the concept will also focus on this area. In skill-based engineering, a standardized automation function can be assigned to each component in form of a skill [19]. A skill is defined as the potential of a production resource to achieve an effect within a domain [11]. When operating a production plant according to the principle of skill-based engineering, knowledge of the individual components and their skills is required. More complex tasks are represented by a composed skill, which consist of several individual skills (basic skills). The creation of composed skills based on basic skills also represents an information aggregation. This use case is intended to show how the introduced concept can be used to create composed skills in a production plant.

### 4.1 Use Case: Fidget Spinner Production

The concept presented in this paper is tested on a production plant for research purposes located at the SmartFactoryOWL. It is a production system consisting of 7 stations that produces fidget spinner. The whole system is implemented using a skill-based engineering concept. Therefore, every component of the system provides standardized skills that can be browsed and executed via an information model in OPC UA. The components have basic skills that provide just simple functions, e.g. move a linear axis. Several of these basic skills are combined to composed skills that represent more complex skills. To create production processes, the order of the execution of composed skills has to be defined, which is more obvious than traditional controller programming. The aim of the work at hand is to create rules that can automatically define composed skills. However, the selection and combination of the composed skills to create a process is not an aim of this work.

The focus of this use case is station 2 of the plant, where the four bearings are mounted. At first, the base body is inserted into the station. To lock the bearings in place, spikes

are mounted on a linear axis move out. Then, the bearings are inserted, too. In the next step, the press moves on a linear axis over the bearings. The press goes down and thus fix the bearings into to body case. Then, the press moves back to the rest position, the spikes move out and the product is transported to the next station. Overall, the pressing process is the core of station 2, since the other steps are related to transportation processes.

## 4.2 Applying Classification

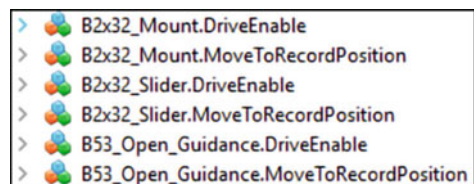
The OPC UA information model reflects the classification of the components. In this use case, it can be seen that there are 3 pneumatic axes. Figure 4 shows a section of the information model. The axes are named *B2x32\_Slider*, *B2x32\_Mount* and *B53\_Open\_Guidance*.

Each axis can move to 2 different positions. According to the classification scheme described in Sect. 3.2, the following taxonomy can be created for each of these 3 axes: The axes are kinematic elements. Since the axes can be actively set in motion, they are active kinematic elements. Furthermore, the axes can be moved along a translatory degree of freedom, so that they are active linear elements. Since the axes are operated pneumatically, the axes are active linear elements with a limit stop. This is the most specific description within the taxonomy. The classifications were taken from the method of the work [6] described in Sect. 3.2. Figure 5 shows a section of this classification.







From the classification of the axes as active linear elements with a limit stop, it follows that each axis has two skills, an enabling skill and one move skill. This represents the mandatory information described in Sect. 3.2, which will be used to define the rules for the *Rule Set*. The two skills are basic skills and are evident from the OPC UA information model (*DriveEnable* and *MoveToRecordPosition*). The move skill needs a parameter that specifies the destination position of the axis. Since it cannot generally be said whether the forward or backward position is the working position, the rest and the working position are defined for every axis. That enables the rules to distinct between these positions for every axis and thus provide more valuable composed skills. Furthermore, the rest position is also the position in which the axis is in a safe position.

The OPC UA information model also shows that two of the pneumatic axes are connected. This means that for a process step both axes must be moved one after the other to the respective working position. The required sequence is not apparent from the

**Fig. 4** Section of the information model of station 2





Symbol	Category	Description
	Kinematics	Is used to lock or unlock the six kinematic degrees of freedom of an object. Active elements can execute a movement in the direction of the supposed degree of freedom.
	Active Kinematic Element	Can actively perform a movement.
	Active Linear Element	Can actively perform a movement along a translatory degree of freedom.
	Active Linear Element with Limit Stop	Can actively perform a linear movement to a defined fixed limit stop.
	Active Linear Element for Positioning	Can actively perform a controlled, linear positioning movement within the working range.
	Active Continuous Linear Element	Can actively perform a continuous linear motion.

**Fig. 5** Classification of a pneumatic axis [6]

information model. This has consequences for the determination of composed skills, which will be described in Sect. 4.3.

### 4.3 Applying the Rule Engine

A *Rule Set* is used to determine which composed skills can be provided by the production system. The *Rule Set* makes use of the classification determined in Sect. 4.2. The classifications result in certain types of components which provide a set of basic skills that can be used to construct composed skills. In this example, two of the pneumatic axes, which are connected to each other (axis *B2x32\_Slider* and axis *B2x32\_Mount*), are considered. However, the sequence in which the axes must be moved in order to execute a production step is not apparent from the OPC UA information model. This means that there are several possibilities in which order the axes can be moved. This issue must be taken into account when creating rules within the *Rule Set*, e.g. by using the experience of experts when creating rules or by excluding meaningless sequences in advance. After the *Rule Set* has identified which composed skills can be created, this information is passed on to the *Mapper*. The *Mapper* ensures that the composed skills are actually generated by creating method calls. The following section shows how the *Rule Engine* can be used to create composed skills for the two connected pneumatic axes.

The following rule can be applied in order to determine a set of basic skills of the axes *B2x32\_Slider* (*X*) and *B2x32\_Mount* (*Y*) and assign a set of composed skills to the production module *Station 2* (*Z*):

$$ALE.LS(X) \wedge ALE.LS(Y) \wedge linked(X, Y) \wedge isPartOf(X, Z) \wedge isPartOf(Y, Z) \\ \rightarrow (hasSkill(Z, CS_z1) \wedge consistsOf(CS_z1, BS_x1, BS_y1)) \wedge (hasSkill(Z, CS_z2) \wedge \\ consistsOf(CS_z2, BS_x2, BS_y2, BS_y1, BS_x1))$$

Table 1 explains the logical expressions which are used in the rule and Table 2 explains the assignment of the variables.

The rule reads as follows. First, it is checked whether the components are from the class *Active Linear Element with Limit Stop* ( $ALE.LS$ ). Then it is checked whether these two components are connected to each other (*linked*). Furthermore, it is checked whether both elements are part of the production module *Station 2* (*isPartOf*). It follows ( $\rightarrow$ ) that several composed skills are added to the production module “Station 2” (*hasSkill*), which make use of the basic skills of the axes (*consistsOf*). The basic skills are known as the expression  $ALE.LS$  queries a certain class, which basic skills can be taken from the classification described in Sects. 3.2 and 4.2. In particular, these are two variants of the skill *MoveToRecordPosition*. The first is used to move the axis to the rest position and the second is used to move the axis to the working position. For simplicity, the basic skill *DriveEnable* is neglected here.

**Table 1** Explanation of logical expressions

Logical expression	Explanation
$ALE.LS(X)$	This expression is true, if the component X is from the category <i>Active Linear Element with Limit Stop</i>
$linked(X, Y)$	This expression is true, if the components X and Y are linked
$isPartOf(X, Z)$	This expression is true, if the component X is part of the station Z
$hasSkill(Z, CSkill)$	This expression is true, if the composed skill “CSkill” is assigned to station Z
$consistsOf(CS_Z, BS_X, BS_Y)$	This expression is true, if the composed skill $CS_Z$ makes use of the basic skills $BS_X$ and $BS_Y$

**Table 2** Assignment of the variables used in the rule

Variable	Value	Description
X	B2x32_Slider	Pneumatic Axis
Y	B2x32_Mount	Pneumatic Axis
Z	Station_2	Station 2
$BS_x1$	$MoveToRecordPosition_S(RestPosition)$	Basic Skill of B2x32_Slider
$BS_x2$	$MoveToRecordPosition_S(WorkingPosition)$	Basic Skill of B2x32_Slider
$BS_y1$	$MoveToRecordPosition_M(RestPosition)$	Basic Skill of B2x32_Mount
$BS_y2$	$MoveToRecordPosition_M(WorkingPosition)$	Basic Skill of B2x32_Mount
$CS_z1$	MoveToSafePosition	Composed Skill of Station 2
$CS_z2$	MountBearing	Composed Skill of Station 2

**Table 3** Available composed skills for the use Case

Composed skill name	Basic skill mapping
MoveToSafePosition	<i>CallSkill(MoveToRecordPosition<sub>S</sub>(Rest Position);</i> <i>CallSkill(MoveToRecordPosition<sub>M</sub>(Rest Position);</i>
MountBearing	<i>CallSkill(MoveToRecordPosition<sub>S</sub>(Working Position);</i> <i>CallSkill(MoveToRecordPosition<sub>M</sub>(Working Position);</i> <i>CallSkill(MoveToRecordPosition<sub>M</sub>(Rest Position);</i> <i>CallSkill(MoveToRecordPosition<sub>S</sub>(Rest Position);</i>

This rule assigns the composed skills to the production module *Station 2* and determines which basic skills can be assigned to the composed skills. The next step is to actually generate the identified composed skills and create corresponding method calls. This is done by the second part of the *Rule Engine*, the *Mapper*. It technically creates the composed skills and the method calls for the basic skills in the sequence, that was identified with the help of the *Rule Set*. Table 3 shows how the basic skills are combined to composed skills.

The *Mapper* stores the created composed skills in a suitable location (e.g. an information model). This ensures, that the system operator is offered the above mentioned composed skills after the *Rule Engine* has finished its process.

## 5 Conclusion and Future Work

In this paper, a concept for rule-based information aggregation in modular production plants is presented. Aggregated information is created from the individual information of a production plant, e.g. from the contents of an information model of a production module. The concept consists of a *Rule Engine*, which determines possible aggregated information with the help of a *Rule Set*. This aggregated information is then created and made available by a *Mapper*. A pre-condition for the aggregation process is that the individual information are available in a standardized form. For this purpose, a classification scheme for production plants was presented, which allows the assignment of plant components into different categories. To test the presented concept, a use case from the field of skill-based engineering was considered. Here the concept was applied to a module of a real production plant. As a result, composed skills were generated from the basic skills of the production plant.

However, this is a first example that shows how the concept works. When applied to more complex systems, it must be ensured that the system complexity is also handled in the *Rule Set*. This means, that the amount of rules required can become very large, so that complexity issues must be taken into account. It is conceivable that different *Rule Sets* have to be created for different aspects of CPPS instead of a global *Rule Set* for the entire plant. However, the creation of *Rule Sets* for different aspects of CPPS is challenging,

since it strongly depends on the given type of information. Also, an essential point of the presented approach is the assumption about the existence of mandatory information in the information models based on standards. If this condition is not met, no rules can be defined, which puts the whole concept into question. This fact could be seen as a weak point. However, since it can be expected that standardization will prevail, the above mentioned assumption is applicable. In future work it has to be shown how the presented concept can be applied to other aspects of CPPS. For this purpose, the concept shall be applied to several different modules of a production plant.

---

## References

1. AutomationML e.V.: AutomationML, <https://www.automationml.org/>
2. Banerjee, S., Großmann, D.: Aggregation of information models – an opc ua based approach to a holistic model of models. In: 2017 4th International Conference on Industrial Engineering and Applications (ICIEA). pp. 296–299 (2017)
3. Bhattacharya, S.: Preference monotonicity and information aggregation in elections. *Econometrica* **81**(3), 1229–1247 (2013)
4. Gilbert, T.: Information aggregation around macroeconomic announcements: Revisions matter. *Journal of Financial Economics* **101**(1), 114–131 (2011)
5. Großmann, D., Bregulla, M., Banerjee, S., Schulz, D., Braun, R.: Opc ua server aggregation – the foundation for an internet of portals. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). pp. 1–6 (2014)
6. Helbig, T.: Methode zur Verbesserung der domänenübergreifenden Zusammenarbeit während des Engineering-Prozesses im Sondermaschinenbau (2016)
7. Helbig, T., Henning, S., Hoos, J.: Efficient engineering in special purpose machinery through automated control code synthesis based on a functional categorisation. In: Proceedings of the 1st Conference on Machine Learning for Cyber Physical Systems and Industry 4.0 (ML4CPS). Lemgo, 01. – 02.10.2015. (2015)
8. Iatrou, C.P., Ketzel, L., Urbas, L., Graube, M., Häfner, M.: Klassifikation und methodischer Entwurf von OPC UA Aggregating Servern für intelligente Architekturen. In: 21. Leitkongress der Mess- und Automatisierungstechnik – Automation 2020). pp. 781–796 (Jul 2020)
9. Kumagai, K., Fujishima, M., Yoneda, H., Chino, S., Ueda, S., Ito, A., Ono, T., Yoshida, H., Machida, H.: KPI element information model (KEI Model) for ISO22400 using OPC UA, FDT, PLCopen and AutomationML. In: 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). pp. 602–604 (Sep 2017)
10. OPC Foundation: Opc unified architecture, <https://opcfoundation.org/about/opc-technologies/opc-ua/>
11. Plattform Industrie 4.0: Glossary, <https://www.plattform-i40.de/PI40/Navigation/DE/Industrie40/Glossar/glossar.html>
12. Ribeiro, L., Björkman, M.: Transitioning from standard automation solutions to cyber-physical production systems: An assessment of critical conceptual and technical challenges. *IEEE Systems Journal* **12**(4), 3816–3827 (Dec 2018)
13. Seilonen, I., Tuovinen, T., Elovaara, J., Tuomi, I., Oksanen, T.: Aggregating opc ua servers for monitoring manufacturing systems and mobile work machines. In: 21st International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–4 (2016)

14. Vinodha, D., Anita, E.A.M.: A survey on privacy preserving data aggregation in wireless sensor networks. In: 2017 International Conference on Information Communication and Embedded Systems (ICICES). pp. 1–6 (Feb 2017)
15. Walzel, H., Vathoopan, M., Zoitl, A., Knoll, A.: An Approach for an Automated Adaption of KPI Ontologies by Reusing Systems Engineering Data. In: 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1693–1696 (Sep 2019)
16. Weyer, S., Schmitt, M., Ohmer, M., Gorecky, D.: Towards Industry 4.0 – Standardization as the crucial challenge for highly modular, multi-vendor production systems. IFAC-PapersOnLine **48**(3), 579–584 (2015), 15th IFAC Symposium on Information Control Problems in Manufacturing
17. Würger, A., Niemann, K., Fay, A.: Concept for an energy data aggregation layer for production sites a combination of automationml and opc ua. In: 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 1051–1055 (2018)
18. Yonghui Shim, Younghan Kim: Data aggregation with multiple sinks in information-centric wireless sensor network. In: The International Conference on Information Networking 2014 (ICOIN2014). pp. 13–17 (Feb 2014)
19. Zimmermann, P., Axmann, E., Brandenbourger, B., Dorofeev, K., A.Mankowski, Zanini, P.: Skill-based engineering and control on field-device-level with opc ua. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1101–1108 (Sep 2019)

**Open Access** Dieses Buch wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Buch enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

