AAS Capability-Based Operation and Engineering of Flexible Production Lines

Yining Huang
CEA-List
Université Paris-Saclay
Palaiseau, France
yining.huang@cea.fr

Saadia Dhouib
CEA-List
Université Paris-Saclay
Palaiseau, France
saadia.dhouib@cea.fr

Jacques Malenfant
Laboratoire d'Informatique de Paris 6 (LIP6)
CNRS, Sorbonne Université
Paris, France
jacques.malenfant@lip6.fr

2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA) | 978-1-7281-2989-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ETFA45728.2021.9613435

***Abstract*—Lot-size-one systems as well as plug and produce concepts imply (1) producing increased variety of products in a highly flexible and timely manner, and (2) making commissioning and maintenance more flexible. The speed with which manufacturers, in particular SMEs, can reconfigure the production to a new run and thus respond to clients and avoid costly machine downtime is critical to maintaining commercial success and profit margins. The manufacturing systems of tomorrow must offer a high degree of autonomy, be quickly re-planned to other operations, and cope with a wide variety of unforeseen situations, in a secure and safe manner. In this context, the Asset Administration Shell (AAS) is an emergent standard that leverages the digital twin approach and provides concepts for describing capabilities and skills of I4.0 components in order to automate the reconfiguration process. This article proposes a capability-based operation and engineering approach to tackle the syntactic and semantic interoperability problems in flexible production lines. We demonstrate the implementation of the AAS standard in the open source model-driven workbench Papyrus; then we assess its usability for modeling a production cell use case in order to implement a capability-based reconfiguration approach for flexible production lines.**

***Index Terms*—Industry 4.0, Interoperability, Digital Twin, Flexible production lines, AAS**

I. Introduction

This paper presents the extension of the model-driven workbench Papyrus to a capability-based engineering (CBE) [1] approach addressing the challenges of Industry 4.0 flexible product lines management. Cast in the lineage of RAMI 4.0 [2] and its Asset Administration Shell (AAS) [3], CBE aims at fully representing in a machine-interpretable form the offered capabilities of resources managed as AAS, hence allowing to automate both the reconfiguration and the operation of flexible product lines. Leveraging the digital twin concept [4] of AAS, CBE proposes to automatically transform series of abstract production workflows (the production lots) exhibiting their required capabilities into production plans that select, configure and operate the resources offering matching capabilities.

Capability-based engineering is a highly challenging goal which raises crucial issues such as the interoperability of capability descriptions as well as the effective monitoring of product lines to ensure the correct execution of production plans. Though the RAMI 4.0 and AAS offer much interoperability by themselves, they do not deal with the representation of capabilities *per se*. In this article, we introduce a capability-based semantic interoperability approach, which is based on referencing clear semantics of an AAS. This approach deals with both the semantics of capability descriptions in the matching of required to offered ones and the semantics of data when matching the monitoring data of the actual product line with its digital twin represented as AAS, which aims at producing the work plan automatically without human intervention.

Next, Section II presents the context and related works. Section III summarizes our approach and focuses on the implementation and usage of the AAS modeling concepts in our Papyrus workbench as well as illustrate our modeling approach through a simple production cell use case. Section IV concludes and discusses future prospects.

II. Background

A. RAMI 4.0 and AAS

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) [2] is the first reference architecture model to accurately describe the Industry 4.0 (I4.0) components. It enables I4.0 stakeholders (architects, developers, business decision makers, etc) to adopt a common perspective and a common understanding of I4.0 systems, resources and assets. RAMI 4.0 encompasses the most important aspects of I4.0 components, making them a worldwide identifiable participant able to communicate, through its AAS (the digital twin part) and an asset (the physical part). The former reflects the asset in the five upper layers of RAMI (integration, communication, information, function and business processes).

The Asset Administration Shell [3] is defined as *a standardized digital representation of the asset. It identifies the Administration Shell and the assets represented by it, holds digital models of various aspects (Submodels in Fig. 1) and describes technical functionality exposed by the Administration Shell or respective assets.* The Submodels are composed of Submodel Elements. These elements can be product properties, process variables and parameters, events for observing properties, references to external data sources or files, capabilities, operations and entities of the composite I4.0 component. The relationships among AAS components are defined in [5], including the composition of entities in a bill of material

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on May 21,2024 at 00:54:11 UTC from IEEE Xplore. Restrictions apply.

Submodel and the property connections established between entities.

B. Interoperability

Interoperability [6] refers to the ability of different systems and organizations to work together seamlessly. It can be subdivided into syntactic interoperability and semantic interoperability. Syntactic interoperability denotes that systems have the ability to communicate and exchange data, which includes specified data formats, communication protocols, interface descriptions, etc. Semantic interoperability goes beyond by considering valid data exchanges between the systems as adequately understood and processed by all parties.

Properties with a standardized meaning provide a common semantic understanding of the meaning of the exchanged data. DIN SPEC 92000 [7] specifies the data exchanges among objects. In this specification, the concept of property value statement (PVS) is proposed. Each PVS is associated with a general property that has a standardized definition (*e.g.*, IEC 61360 [8], ecl@ss [9]), which fixes the semantics of the property. With the concepts of AAS and capability-based engineering [1], one can integrate all the information of a digital factory. So the main goal is to achieve the alignment between the assured capabilities provided by the production resources and the requirements from the production process. In our approach, we will consider the DIN SPEC 92000 as a candidate to describe the semantics of provided or required capabilities of an AAS component.

C. Related work

A detailed overview of RAMI 4.0 and AAS is given in [10], and [11] provides an in-depth explanation of the concepts defined in AAS and gives suggestions to Industry 4.0 stakeholders. Open source AAS implementations (e.g. BaSyx [12], NovAAS [13]) provided by different organizations are discussed in [14]. NovAAS [13] contributes towards a web-based reference implementation of the concept asset administration shell. In [15], the concepts of property value statement are presented which fit to the standardized property definition models and build a platform for a metadata-based information handling in the digital factory. A. Perzylo et al. [16], introduces concepts developed by the BaSys 4.0 initiative dealing with the semantics of manufacturing skills, orchestrating higher-level skills from basic skills, and using them in a cognitive manufacturing framework. Now researchers tend to integrate OPC UA Information Models to the AAS skills descriptions; [17] shows a mapping between AAS concepts and OPC UA information models.

Our approach aims at introducing a capability-based semantic interoperability approach, which is based on referencing clear semantics for each element (submodel, submodelElement) of an AAS. Semantics can be defined in an ontology or in a catalog like ecl@ss [9].

III. Capability-based engineering approach & AAS Modeling concepts for CBE

A. Capability-based engineering approach

In capability-based engineering captured from [1], the capability concept is a abstract description of the functionality of a production resource while the skill concept is the asset-dependent implementation to achieve a certain effect. Capability descriptions are meant to simplify the task of maintenance operators to match equipment with similar functions as a defected one. The main goal of capability-based engineering is to design, implement and then dynamically operate the system according to the functions required in each step of the production process, rather than explicitly specifying the actual production resources.

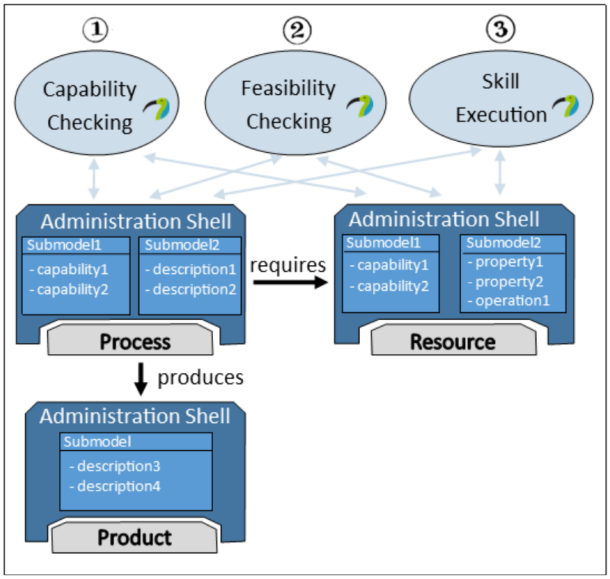

Fig. 1. Capability-based engineering approach for flexible production lines

Resource, process and product, these three elements play key roles in order to achieve capability-based operation and engineering. The AAS of a resource describes the provided capabilities and related features, without knowing the process to complete and the product to produce. The AAS of a process describes the capabilities required by the work plan and some environmental constraints. The AAS of a product describes the product from different aspects. Flexible production systems must automatically match resources, processes and products in order to achieve continuous capacity-based engineering and minimize production line downtime. Our approach (Fig. 1) tackles this in three main steps:

- Capability Checking: a matching of the capabilities offered by the resources with the capabilities required by the production process, for each resource used to produce a certain product. One feasible method is to define a semantic description language using Web Ontology Language (OWL). OWL concepts can be connected and combined using semantic rules, so reasoners will

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on May 21,2024 at 00:54:11 UTC from IEEE Xplore. Restrictions apply.

infer possible solutions for required/provided capabilities matching.

- Feasibility Checking: a procedure to bring in environmental factors and constraints, and perform simulations for feasibility checks of the solutions found during the capability checking phase. This will further filter out the skills which meet the conditions and will be deployed on the production line.
- Skill Execution: puts the resources into operation according to the process work plan. In this phase, a middleware is needed to automatically perform the skills execution. BaSyx [12] is an appropriate candidate for the skill execution.

Thanks to the automation of phases (1, 2, 3) in Fig. 1 the work plan can be quickly produced just before execution which currently require a lot of human intervention. In the pre-production stage, capability checking and feasibility checking will be completed. At the end of this phase, we will get a production line reconfiguration plan for skill execution. In the rest of the paper, we demonstrate the AAS modeling concepts necessary for the realization of the aforedescribed steps of the approach. We focus on: (1) How to model AASs, their capabilities and skills, (2) How to model composition between AASs and (3) How to describe a production process as an AAS. The actual implementation of capability checking, feasibility checking and skill execution will be the focus of our future work.

B. AAS Modeling environment in Papyrus

Our AAS graphical modeling environment is an extension of Papyrus[1]. We implement the AAS meta-model defined in the specification [3] as a UML profile, which allows I4.0 systems to be modeled in conformance with the AAS standard. Based on the Papyrus UML modeling framework, we developed a graphical modeling environment that customizes UML class diagrams, UML composite structure diagrams and BPMN[2] process diagrams (an extension of UML activity diagram). BPMN is a standardized business process model notation; reusing it avoids the confusion that might be caused by redefining meta-models. This modeling environment provides a simple way to create AAS without the need for an in-depth understanding of the AAS specification. With this environment, not only can the structure of the AAS model be created and visualized, but also the production process can be designed.

The next sections illustrate the AAS modeling concepts necessary to apply capability-based engineering on a simple use case. The chosen use case is a production cell composed of: (1) production resources: a robot, a drilling machine and a coloring system, (2) a production process, and (3) a product.

C. AAS, capabilities and skills Modeling

We use AASs to describe all the components that appear in this example, the process, the product, the production cell and

[1]https://www.eclipse.org/papyrus/

[2]https://www.omg.org/spec/BPMN/2.0/About-BPMN/

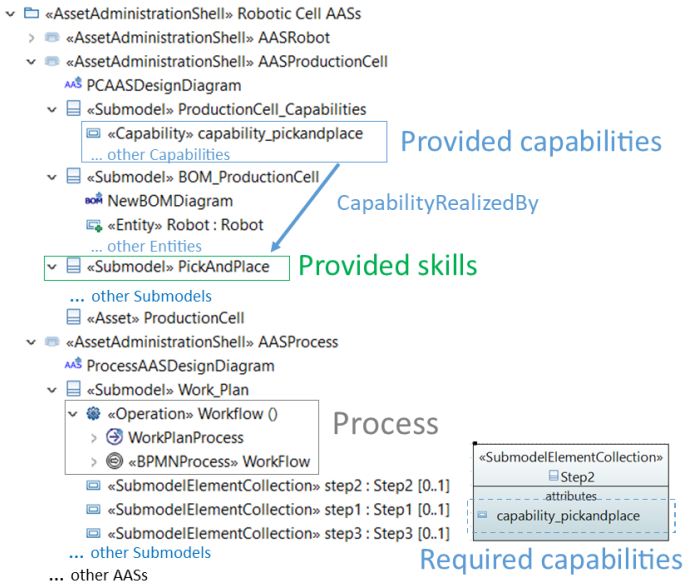

Fig. 2. Structure of an Asset Administration Shell Package

the sub-systems. AAS diagrams, an extension of UML class diagram, can be created in Papyrus to model the structure and relationship within an AAS. By definition, an AAS consists of one or more submodels which describe the referenced asset by different aspects. Moreover, the submodels contains various kinds of submodel elements (property, capability, operation, entity, etc.) in order to fully represent the abstract features of the asset. As shown in Fig. 2, the production unit has a submodel that describes the capabilities, a functional submodel for the implementation of the capabilities, also a composition submodel that shows the internal structure of this asset. The submodel element “Capability” can be annotated with qualifiers. Property value statements (introduced in II-B), a type of qualifier, can be used to describe the capability’s kind (provided or required). PVS consist of several basic elements, namely the ID of the PVS, expression semantic, related property, related subject, predicate value and predicate relation. The expression semantics classifies the meaning of the logic expressed with the predicate, possible expression semantics are ”requirement”, ”assurance”, ”measurement”, ”offer” etc. The ”requirement” represents the capability *required* by the process. Both ”assurance” and ”offer” express the capability *provided* by the resource. However, the ”assurance” represents rather the resources that have passed the feasibility checking, while ”offer” is independent with other capabilities or execution context.

D. Process Modeling

The production process can also be represented in an Asset Administration Shell with a submodel ”Work Plan” where we can define the process steps and requirements. We choose to attach the process model to the submodel element ”operation” as shown in Fig. 2, from this submodel element a BPMN Process diagram can be created to describe the tasks and possibly

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on May 21,2024 at 00:54:11 UTC from IEEE Xplore. Restrictions apply.

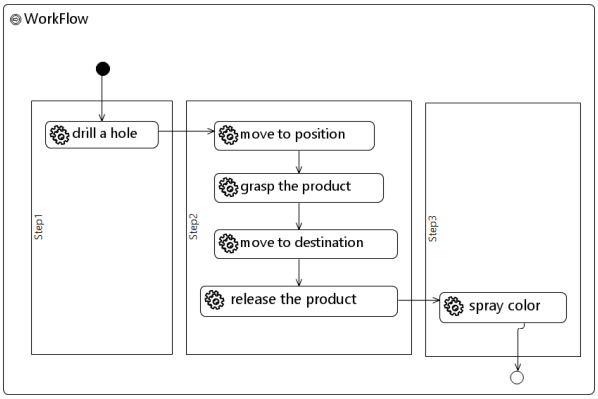

Fig. 3. Workflow BPMN Process diagram

execute them. This is a customized UML activity diagram composed of a subset of BPMN concepts as mentioned in III-B. The scenario defined in the example is that we want to first drill a hole in the raw material, then pick up the material by a robot arm and place it onto the coloring system and finally complete the coloring action. The process diagram (Fig. 3) shows three different steps referring to three different systems in this production cell.

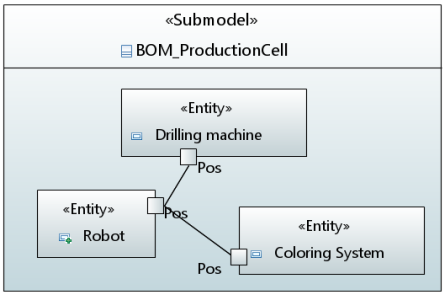

Fig. 4. Bill of material composition diagram

E. Composite AAS Modeling

I4.0 components represented by an AAS can be combined into a new I4.0 component [5]. Then certainly a composite I4.0 component should provide a composite method to show the sub-components it contains. By defining a bill of material (BOM) in a Submodel, all the sub-components are listed here as "Entities" (see Fig. 2 for Robot, Drilling Machine & Coloring System). These entities can be classified into two types, co-managed and self-managed. Co-managed entity means that the entity needs to be managed by a higher-level entity, because it does not have its own asset administration shell. On the contrary, a Self-managed entity has an AAS attached to it. The connections between entities are realized by connecting their properties which are defined by statements (more details in [3], [5]). In order to visualize the composite AAS and the connections between sub-components, it is possible to create a BOM diagram (Fig. 4) for the BOM Submodel of the composite AAS. For example, the BOM diagram of the production cell contains three entities, the robot, the drilling machine and the coloring system. These entities are related with each other by the common property "Pos" (for position).

IV. Conclusion and Future Work

In this paper, we have shown that our model-driven engineering workbench Papyrus is very well-suited to derive a capability-based engineering workbench that is fully compliant with the standards RAMI 4.0 and AAS. The capability-based engineering methodology we implemented fosters the interoperability between I4.0 components as a key enabler to achieve flexible production. As presented above, the capability-based continuous operation and engineering foresee the three different phases: capability checking, feasibility checking and skill execution. In the future, continuous work will be carried out on the automated production process part of this engineering tool. We plan this orchestrating part to be able to automatically complete the three key steps of capability checking, feasibility checking and skill execution.

References

[1] "Describing Capabilities of Industrie 4.0 Components," , Plattform Industrie 4.0 , December 2020. https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Capabilities_Industrie40_Components.html
[2] R. Heidel, M. Hoffmeister, M. Hankel, and U. Döbrich, "The Reference Architecture Model RAMI 4.0 and the Industrie 4.0 component," 2019, VII, 150 pages, Din A5, Broschur.
[3] "Details of the Asset Administration Shell - Part 1", Plattform Industrie 4.0 , November 2020. https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html
[4] E. Glaessgen and D. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," AIAA, Honolulu, Hawaii, Apr. 2012.
[5] "Relationships between I4.0 Components – Composite Components and Smart Production. ," Plattform Industrie 4.0 , June 2017. https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/hm-2018-relationship.html
[6] A. Zeid, S. Sundaram, M. Moghaddam, S. Kamarthi, and T. Marion, "Interoperability in Smart Manufacturing: Research Challenges," Machines, vol. 7, no. 2, p. 21, Apr. 2019.
[7] DIN SPEC 92000:2019-09 "Data Exchange on the Base of Property Value Statements (PVSX)", 2019 September.
[8] Common Data Dictionary (CDD—V2.0014.0013), IEC 61360-CDD.
[9] eclass. Aug. 2017. [Online]. Available: http://www.eclass.eu/
[10] X. Ye and S. H. Hong, "Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells," IEEE Industrial Electronics Magazine, vol. 13, no. 1, pp. 13–25, Mar. 2019.
[11] C. Wagner et al., "The role of the Industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," ETFA, Sep. 2017.
[12] "Bundesministerium für Bildung und Forschung", Basys 4.0, [online] Available: https://www.basys40.de/.
[13] G. di Orio, P. Maló and J. Barata, "NOVAAS: A Reference Implementation of Industrie4.0 Asset Administration Shell with best-of-breed practices from IT engineering," IECON 2019, pp. 5505-5512.
[14] E. Barnstedt et al., "Open Source Drives Digital Twin Adoption," Mar. 2021.
[15] U. Epple, M. Mertens, F. Palm and M. Azarmipour, "Using Properties as a Semantic Base for Interoperability," in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3411-3419, Dec. 2017.
[16] A. Perzylo et al., "Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective," IFAC-PapersOnLine, vol. 52, no. 13, pp. 1590–1596, 2019.
[17] J. Fuchs, J. Schmidt, J. Franke, K. Rehman, M. Sauer, and S. Karnouskos, "I4.0-compliant integration of assets utilizing the Asset Administration Shell," ETFA, Zaragoza, Spain, Sep. 2019, pp. 1243–1247.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on May 21,2024 at 00:54:11 UTC from IEEE Xplore. Restrictions apply.