Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 109 (2022) 227–232

www.elsevier.com/locate/procedia

32nd CIRP Design Conference

Integration of Communication using OPC UA in MBSE for the Development of Cyber-Physical Systems

Johannes Olbort [a], Benjamin Röhm [a]*, Vladimir Kutscher[a], Reiner Anderl[a]

[a]Department of Computer Integrated Design, Technical University of Darmstadt, Otto-Berndt-Strasse 2, Darmstadt, 64287, Germany

* Corresponding author. Tel.: +49-6151-16-21847; fax: +49-6151-16-21843. E-mail address: roehm@dik.tu-darmstadt.de

Abstract

Interoperability of Cyber-Physical Systems in the context of Industry 4.0 is currently not explicitly considered during product development leading to compatibility problems during commissioning and usage. This paper addresses the research question of integrating interoperability based on OPC UA Companion Specifications and its building blocks into Model-Based Sytems Engineering (MBSE). For this purpose, a system modeled in SysML is extended by the requirements resulting from communication via OPC UA. The generated information model is integrated into the OPC UA server, enabling interoperability between Cyber-Physical Systems. A prototypical implementation validates the approach.

Keywords: Interoperability, OPC UA, MBSE, CPS

1. Introduction

Due to the ongoing digitalization, a change in industrial production is taking place, which creates opportunities and challenges for manufacturers and developers. The networking of physical real and virtual systems to cyber-physical systems (CPS) offers the possibility of increasing competitiveness by increasing flexibility. They form autonomous production units, configure and optimize themselves and are expandable without manual setup. Business models and processes are therefore undergoing major changes in order to become more effective and efficient [1].

The networking and communication of machines and systems with each other is, therefore, a central challenge [2]. Technologies such as Cyber-Physical Systems and especially communication technologies such as the Open Platform Communication Unified Architecture (OPC UA) developed by the OPC Foundation are becoming increasingly important. In order to realize the vision of Industry 4.0 of a networked, intelligent world [2], however, this development often takes place as further advancement of existing machines and systems. After the actual product development and manufacturing, it is checked which information and interfaces are required to operate the product in a communication-capable manner. The suitability regarding communication of a product in use is not considered during its development in contrast to many other specifications from later product life phases of the product life cycle, as for instance, cost efficiency [3].

2. State of the Art

This chapter provides an overview of the current state of the art. First, an introduction to the Industry 4.0 is given, followed by an introduction to the service-oriented communication technology (OPC UA) used in this work, and finally, a discussion of system development in the context of Model-Based Systems Engineering (MBSE).

10.1016/j.procir.2022.05.241

2.1. Industry 4.0 and IoT

The term Industry 4.0 originates from the project of the same name, which is funded by the Federal Ministry of Education and Research through the Research Union Economy-Science. In the final report of 2013, the aim was to leverage existing technology and market potentials to ensure the competitiveness of the German industry in the future [2].

In Industry 4.0, a new level of organization and management of the entire value chain over the life cycle of products is aimed. To achieve this, all relevant information must be available in real-time by networking all the instances involved in the value chain [4]. This leads to the Internet of Things (IoT), defined as a network of Industry 4.0 components that are connected [5]. This can be a production system, a single machine or just an assembly within a machine. In order to provide properties, it must be communication-capable and needs a virtual representation with a directory of data contents. This virtual representation can contain data covering individual phases of the life cycle, such as manuals, or technical functionality, such as software for configuration or operation. Such a component is a specialization of a Cyber-Physical System (CPS). [4]

Cyber-Physical Systems (CPS) are considered the technological basis for implementing the Industry 4.0 and IoT projects. They describe a combination of physical (real) objects, such as systems with sensors and actuators, with information-processing (virtual) objects. CPS can be created by extending mechatronic systems with a communication interface. [6–8]

2.2. OPC UA

OPC UA is a service-oriented communication architecture developed by the OPC Foundation. It represents an extension of the previous OPC industry standard by important functions such as platform independence, scalability and Internet capability and is seen as a common intersection to fulfill the requirements of machine-to-machine (M2M) communication, the Internet of Things (IoT) and Industry 4.0. The integration capability of OPC UA across all levels also ensures communication horizontally between systems and vertically between systems and applications at higher enterprise levels. [9]

In the OPC UA architecture, complex information is represented by nodes, which are related to each other by means of references. This is done in the address space. A node can either be instantiated as an object, a variable, a method or a view or define an object type, a variable type, a data type or a reference as a type definition. In either case, a node also has several attributes. These always include a node ID (NodeID). It allows the node to be uniquely identified.

Furthermore, each node has a non-localized name (BrowseName). Depending on what the node represents, further attributes can be added therefore, variables always have an attribute with a data type. It is important to note that the OPC-UA architecture only describes how clients access information in the server, not how this information is to be organized [9]. Such a description takes place in the information models.

2.3. Model-Based Systems Engineering

Model-based systems engineering (MBSE) is presented by the International Council on Systems Engineering (INCOSE) as an interdisciplinary approach to complexity management in product development [10]. The principles of Systems Engineering (SE) are applied to a central model. In contrast to classical document-based development, where documents are created and exchanged between project participants, all necessary data is brought together in this model. This results in advantages, such as the constant actuality of data, since it can be accessed in real-time. The data is also machine-readable and can be further processed, if necessary, with the help of tools. In addition, further machine-readable data, such as code or files in XML format, but also human-readable data such as documents and lists, can be generated. [11, 12]

Model-based system engineering uses a modeling language for syntax and semantics. In this context, the Systems Modeling Language (SysML) developed by the Object Management Group (OMG) deserves special mention. It is based on Version 2.0 of the Unified Modeling Language (UML), which was further improved in 2007 and was developed and designed in cooperation with INCOSE in order to increase the degree of abstraction of programs and thus to represent logical sequences of programs independently of the programming language [13]. The basic principle of UML is to abstract systems by means of models and diagrams as information carriers and, under certain circumstances, to display them from different angles. These are then coupled with each other to obtain a holistic picture of the system. In order to maintain the clarity of complex systems and, therefore, the benefits of a model-based representation, UML also offers the possibility of creating profiles and extending model elements by so-called stereotypes. They allow adaptation to domain-specific problems, for example, using additional properties or conditions. [14]

3. Concept

This chapter introduces the concept of integrating communication via OPC UA in model-based systems engineering. The concept consists of four steps, which are explained after the concept overview in figure 1 within in a SADT-Diagram. The Structured Analysis and Design Technique (SADT) is used to design the process and to derive the information model according to the Unified Modeling Language (UML) [15]. The diagram is composed of four activities. Each processes the incoming input-data (arrow from the left) according to the control data (arrow from above) using the corresponding resources (arrow from below) to produce the output data (arrow to the right).

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

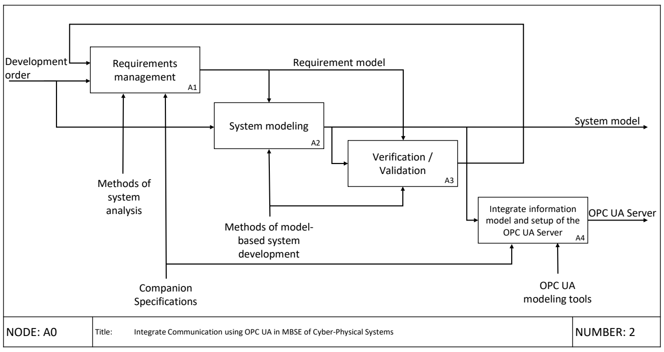

Fig.1. Concept overview in SADT-Notation

The use of Companion Specifications ensures a structured, semantic representation of the data, which makes producer-independent information exchange possible. As shown in Figure 1, a system model (A2) and an OPC UA server (A4), whose information models are enriched for the industry 4.0 communication, will be created based on a development order. Such a server with its information models represents the functional center for later communication. The four activities of the concept: *Requirements management*, *System modeling*, *Verification / Validation*, *Integrate information model and setup of the OPC UA Server* are explained below.

3.1. Requirements Management

The goal of *requirements management* is to handle all requirements. This is to avoid inconsistencies between product requirements, project planning and work resources [16]. In addition to the existence of communication technology, further requirements are made. These occur in the form of non-functional requirements and framework conditions, such as required security protocols or the hierarchical levels to be covered within factories, plants or companies. However, since this concept makes use of existing communication technology in the form of OPC UA, these are decisive for the possible selection of communication technology, but not for the product development itself and therefore not adopted as requirements.

This makes it more important to ensure compliance with the information modeling specified by OPC UA in the form of Companion Specifications (CS). They specify which information is presented and how, in order to guarantee the conformity of systems within an industry or, in the case of OPC UA for Machinery, for example, cross-industry interoperability. Therefore, suitable CS is selected based on the planned application area of the system to be developed. Their defined building blocks of the later architecture of the information models are then transferred as requirements regarding a product's communication capability. If these requirements should not cover all desired functions of the system, then these are consulted in the requirement analysis for an extended information model.

3.2. System modeling and Verification / Validation

In the *system modeling,* the data provision interfaces of the system are determined. It is possible, for example, that data can be made available from sensors of the system or can be taken from the control of the system. For communication via OPC UA, however, these must only be accessible via a communication interface. How exactly this is implemented is not important. In addition to the possibility of first extracting the data from the data sources with an OPC UA client and then sharing it with a central OPC UA server, it is also possible to make the data sources themselves capable of communication. For many devices or programmable logic controllers, adapters are available for upgrading OPC UA functionality [17].

In a model representation of a technical-physical architecture, software parts or components cannot be represented, but they do play an important role in data availability. Therefore, using the so-called functional architecture of system development is suitable for modeling. In the chain of effects architecture, components of a system are represented independently of whether it concerns hardware or software, as functional architecture in the functional group. The focus is on the functional mode of operation of the components of a system. Thus components, which do not make a direct functional contribution, do not appear as, for instance, boards or microprocessors in this architecture representation, although they are necessary for the entire system.

In addition to the necessary development steps in which the product is designed and specified, *verification and validation* steps must always be carried out during the development of a system. It is checked both during development whether the system requirements are implemented by the design and after implementation with the help of appropriate tests to determine whether the system works properly. The verification of the development is carried out, for example, by reviews of project participants, but tools can also be used to check compliance with certain modeling rules and specifications [14]. If weaknesses are identified, both the requirements and the system architecture or specification must be reviewed and revised if necessary.

3.3. Integrate information model and setup of the OPC UA Server

In the last step of the concept, an OPC UA Server is configured to serve as the communication interface of the system. In order to avoid the need to set it up manually, one or more information models for the server are derived from the system model. Several information models are created if the system model is based on several Companion Specifications. For this purpose, it is referred again to requirement management. Over the <<*Satisfy*>> relationship between elements in the requirement diagram, it is indicated which requirements by certain system components are fulfilled. In addition, building blocks are identified, which must be present in the information model. If these building blocks are static variables, these can be initiated directly with a value. A requirement can also be fulfilled by several elements. The initialization of the OPC UA Server is finally done by importing the information models gained in this way, together with the Companion Specifications used in the creation of the system model. The CS must be additionally present since they define the object and variable types, which the information models instantiate.

4. Implementation and Validation

After the concept was presented in the previous chapter, a description of the prototypical implementation of the concept follows. A laser plotter from the Department of Computer Integrated Design is used as a validation object. Besides the presentation of the laser plotter, the corresponding system model is discussed. Afterward, the last steps of the guide are completed.

4.1. Implementation

The laser plotter used as a validation object is a modified version of the "XY Plotter Robot Kit" from Makeblock Co. It has two process axes, which are adjusted by stepper motors and cover a working area of 300 mm by 350 mm. The system is controlled by a Raspberry Pi 3 Model B+ from the Raspberry Pi Foundation. The laser plotter is an already developed production machine. However, the concept is aimed at systems to be developed. Since no system model existed, a laser plotter model was first designed within the Enterprise Architect from Sparx Systems. This model reflects the current state of development and will serve as an entry point for possible further development as a variant or new version. The system model of the laser plotter is explained by using the functional architecture in the appendix A.

Figure 3 in the appendix shows the functional architecture of the laser plotter. Two inputs and two outputs were identified. Inputs of the system are, on the one hand, the power supply by means of electricity to supply the system components, and on the other hand, orders by the user. These are performed by the input of vector graphics. The outputs are, firstly, the information flow in the form of optical signals via the graphical user interface and secondly, the drawing produced by the plotter. The model can be divided into three subsystems human-machine interface, control and mechanics.

The human-machine interface of the laser plotter accepts inputs and orders from the user and passes them on to the control system. Orders are accepted in the form of vector graphics and translated into G-code before they are passed on to the control system. The user can receive feedback via the graphical user interface. The user's order translated into G-code is processed in the controller. This is done by the controller going through the code instruction by instruction and converting it into control currents for the motors. This describes both where and how fast they are moving and the path they have to follow. The mechanical subsystem primarily describes the stepper motors and the axes driven by them via belts on which the tool head can move. The subsystems presented here cannot be transferred one-to-one to the physical components of the system. The Raspberry Pi, for example, provides the human-machine interface as well as parts of the control system.

The laser plotter already meets the basic requirements of a CPS, as the Raspberry Pi has a network interface. Via this interface, a connection to an IP-based network can be established using either WLAN or Ethernet cable, in which an OPC UA server can also be set up. Therefore, the requirements management can be started directly. In addition to the design of the OPC UA for Machinery, the CS OPC UA for Machine Tools serves as the basis for the integration of communication during the development of the CNC laser plotter. In order to be able to use the building blocks contained in the CS in MBSE, an import via the interfaces offered by modeling tools is desired.

Companion Specifications can, just like other OPC UA information models, be integrated into an OPC UA server by importing the corresponding NodeSet, which is available as an XML file. This eliminates the need for time-consuming, manual programming of OPC UA objects, variables and methods.

Figure 2 shows a requirements diagram for the information blocks contained in the CS *OPC UA for Machinery*. The three requirements *ProductInstanceUri*, *SerialNumber* and *Manufacturer,* are listed as mandatory, which means that every machine and system must have at least this information. As specified in the CS, they must be integrated into the architecture of a server so that the system complies with the CS *OPC UA for Machinery*. In case several systems are connected

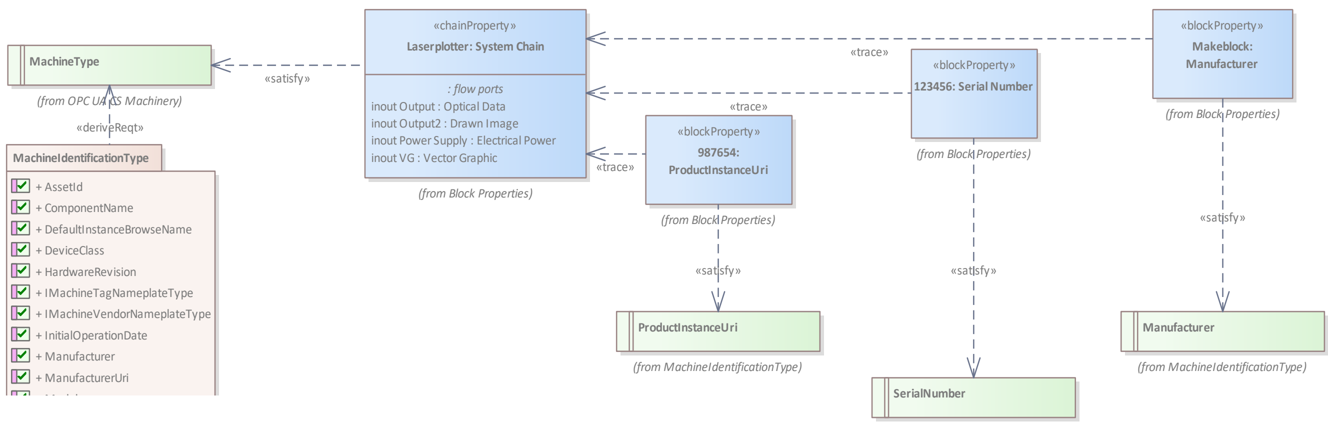

Fig. 2: Requirements Diagram OPC UA Machinery

to one server, the server must be able to distinguish between the information of the individual systems.

4.2. Validation

The concept for the integration of communication via OPC UA in MBSE was able to be implemented by the usage of a CNC laser plotter as a reference production machine. The sequence of steps described in it ensured a structured approach. By choosing the Companion Specifications depending on the application area of the system, it allows applicability within any industry. Even if no industry standard models are (yet) available for these, the concept allows a procedure by considering extended information models in the requirements analysis. Thanks to the use of OPC UA and its modular structure, the system's complexity is also manageable in any case since the implementation of several information models per server is just as realizable as setting up several servers.

Since the concept's requirements management transfers the building blocks defined by Companion Specifications as requirements for communication capability, the concept not only takes existing industry standards into account but also enforces them. During implementation, it was also possible to import explanations for requirements in the case of CS *OPC UA for Machinery*. In CS *OPC UA for Machine Tools* these are not stored in the NodeSet but are available in the accompanying document only. Consequently, a transfer of explanations, modeling rules and data types can only be partially fulfilled. At this point in time, the Companion Specifications are already released, while some are only early drafts and will be extended by further functionalities in the future. As soon as this has been done, there is nothing to prevent a new transfer since the developed interfaces were not designed information model specific. Even in the case of an extension of the laser plotter, which calls for a re-evaluation of the selected Companion Specification used, the concept makes it possible to easily add further industry technologies and integrate them into the system model. In summary, the concept implements an integration of communication through OPC UA in the development context of model-based system engineering.

5. Outlook

The paper showed that the integration of communication via OPC UA in MBSE is possible. The subject of this chapter is the practical suitability and possible extensibility of the concept. Most CS are currently still in development and are not available or only as a draft. In the future, CS for several industries will be available. Of the CS used in this work, the first release version of OPC UA for Machine Tools appeared just about a month ago, at the end of September 2020, and OPC UA for Machinery is currently only available as a draft. Based on those CS, it could be shown that new systems can be developed with communication suitability.

The concept still offers a lot of potential for further development. For example, the translation of CS into importable data is automated, but not automatically. In the beginning, the root node still has to be identified and selected manually, and for the import into a SysML modeling tool, adherence to folder structures is essential. Also, up to now, it is only possible to obtain information from the NodeSet that maps the respective Companion Specification. An object type or a variable, which refers to information from other information models, cannot be specified in such a case. In the concept, this is shown, for example, by the fact that many nodes have only i=68 as object type because the assignment to the PropertyType from the NodeSet, which implements the basic functionalities for OPC UA, is not repeated in the respective CS. A universal database from which one can select the desired Companion Specifications for a to be developed system could solve this issue. This can be expected due to the ongoing digitalization, especially in the industrial environment, and the importance of systems ready for Industry 4.0 communication. Further functions provided by OPC UA, such as the subscription of server-side events by alarms and conditions of clients, could be integrated into the concept. Also, methods that can be implemented on the server have not been applied within the concept so far.

Acknowledgement

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the "Innovations for Tomorrow's Production, Services, and Work" Program (02J19B012) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

Appendix: Functional Architecture

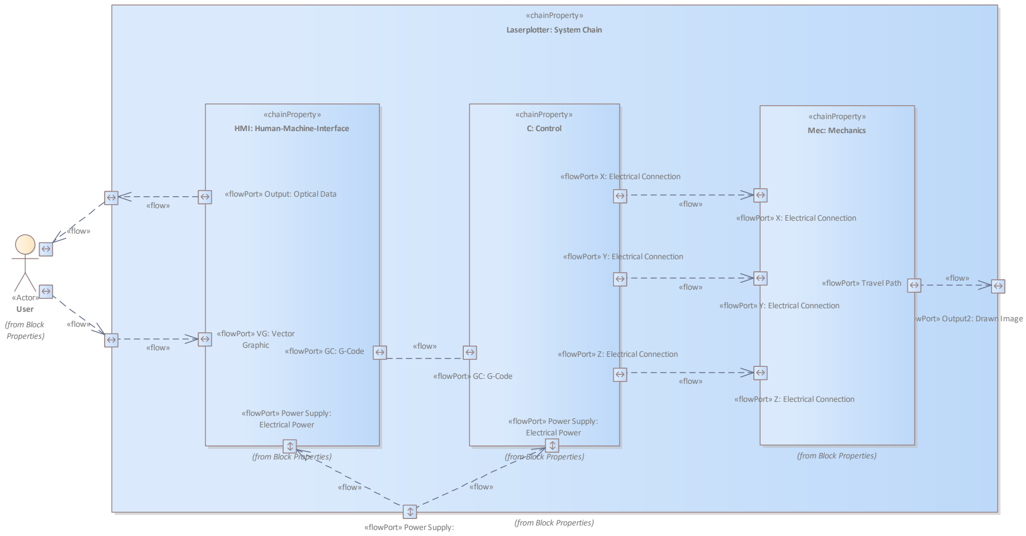

Fig. 3 Functional Architecture of the CPS

References

[1] Abele E, Anderl R, Metternich J, Arndt A, Wank A. Industrie 4.0 - Potentiale, Nutzen und Good-Practice-Beispiele für die hessische Industrie. Studie: Zwischenbericht zum Projekt Effiziente Fabrik 4.0. Meisenbach GmbH Verlag, Bamberg 2015.

[2] Kagermann H, Wahlster W, Helbig J. Recommendations for implementing the strategic initiative Industrie 4.0. Final report of the Industrie 4.0 Working Group 2013.

[3] Wartzack S, Meerkamm H, Bauer S, Krehmer H, Stockinger A, Walter M, Schleich B. Design for X (DFX), 2nd edn., pp. 463–484. Carl Hanser Verlag GmbH & Co. KG, München Hanser.

[4] BITKOM e.V., VDMA e.V., ZVEI e.V.. Implementation Strategy Industrie 4.0. Report on the results of the Industrie 4.0 Platform. 2016.

[5] Alexopoulos K, Koukas S, Boli N, Mourtzis D. Architecture and development of an Industrial Internet of Things framework for realizing services in Industrial Product Service Systems. Procedia CIRP 2018, 880–885.

[6] VDI-/VDE-Gesellschaft Mess- und Automatisiernungstechnik (GMA) (ed.), Cyber-physical systems: Chancen und nutzen aus Sicht der Automation, Thesen und Handlungsfelder (2013).

[7] Lee E A, CPS foundations, Anaheim, California, p. 737. ACM Press, New York, USA 2010.

[8] Kutscher V, Olbort J, Anokhin O, Bambach L, Anderl R. Upgrading of Legcy Systems to Cyber-Physical Systems, Dublin, Ireland 2020.

[9] OPC Foundation, OPC Unfied Architecture. Interoperability for Industrie 4.0 and the Internet of Things 2020.

[10] Katzwinkel T, Jacobs G, Löwer M, Konrad C, Sulejmani F. Beiträge und Herausforderungen modellbasierter Systementwicklung in frühen Phasen des PEP aus Sicht der Forschung. 15. Gemeinsames Kolloquium Konstruktionstechnik 2017 Interdisziplinäre Produktentwicklung 2017.

[11] Alt O. Modellbasiertes Systems Engineering und seine Technologien als Schlüssel für Industrie 4.0, Hanser, München 2015. p. 1–9.

[12] Maccalman A, Lesinski G, Goerger S. Integrating External Simulations Within the Model-Based Systems Engineering Approach Using Statistical Metamodels. Procedia Computer Science 2016.

[13] Weilkiens T, Soley R.M. Systems Engineering mit SysML/UML. Anforderungen, Analyse, Architektur. dpunkt.verl., Heidelberg 2014.

[14] Alt O. Modellbasierte Systementwicklung mit SysML. Carl Hanser Fachbuchverlag, München 2012.

[15] Filippi S. The Design Guidelines Collaborative Framework. Springer, Dordrecht 2009.

[16] Maltzahn S. Simultane Anforderungs- und Architekturdefinition auf Basis semantischer Ähnlichkeit von Anforderungen. Forschungsberichte aus dem Fachgebiet Datenverarbeitung in der Konstruktion. Shaker, Aachen 2013.

[17] Schleipen M, Mersch H, Aro J. Praxishandbuch OPC UA. Grundlagen, Implementierung, Nachrüstung, Praxisbeispiele. 2020.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.