ARTICLE TEMPLATE

An AutomationML extension towards interoperability of 3D virtual commissioning software applications

Jiaqi Zhao[a,b], El-Houssaine Aghezzaf[a,b] and Johannes Cottyn[a,b]

[a]Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 46, Gent-Zwijnaarde, 9052, Belgium; [b]Industrial Systems Engineering (ISyE), Flanders Make, www.FlandersMake.be, Belgium

ARTICLE HISTORY

Compiled October 10, 2023

ABSTRACT

To achieve interoperability between different 3D virtual commissioning software, a generic virtual commissioning data model is required. AutomationML is a standard neutral format for interoperability in the engineering phase. However, the current AutomationML standard is not sufficient for full-scope 3D-based virtual commissioning data exchange, as attributes and modeling method of 3D virtual commissioning-related sensors, actuators and signal connections are not standardized in AutomationML. To fill this gap, the authors suggest extending AutomationML for interoperability in 3D virtual commissioning. In this paper, a case-driven iterative approach is introduced to evolve towards an AutomationML extension. This extension is gradually developed by taking the union of all virtual commissioning-related functions and attributes of 3D virtual commissioning software. During the iteration, naming rules are applied when a new attribute is added to the extension. With this approach, an initial AutomationML extension is created by implementing a first iteration. The interoperability performance of this extension is subsequently evaluated by conducting data exchange of a representative set of 3D emulation models between two 3D virtual commissioning software, namely Siemens NX and Visual Components, via self-developed “Import” and “Export” plug-ins. It shows that AutomationML extension-based data exchange converts 70% more attributes than that only based on AutomationML.

KEYWORDS

Virtual commissioning; AutomationML; Generic data model; Data exchange; Interoperability; Case-driven iterative approach

1. Introduction

Digital Twin (DT) is gaining increasing attention in Industry 4.0 (Tao et al. 2018; Lattanzi et al. 2021), and Virtual Commissioning (VC) is crucial in building DT systems (Barbieri et al. 2021). With VC, cost and time in real commissioning will be significantly reduced, since errors can be detected and modified during the emulation phase (Metzner et al. 2022). Due to these advantages, various 3D virtual model-based VC (3D VC) software applications have been developed (Hoffmann et al. 2010). As each of these software applications has its own specific functionalities, to make the best performance of VC, more than one 3D VC software might be used in industrial practice.

CONTACT Jiaqi Zhao. Email: Jiaqi.Zhao@UGent.be

This fuels the need for interoperability between different 3D VC software applications. However, interoperability between 3D VC software applications is still challenging as each software is developed in compliance with its own principle. In this respect, no common standard has been published yet for describing 3D VC data comprehensively and accurately (Ugarte et al. 2022). This builds a big obstacle for different 3D VC software end users to collaborate, as 3D emulation models must be created manually in each 3D VC software application, which is extremely time-consuming (Thongnuch, Fay, and Drath 2018).

Although for some 3D VC software applications, 3D Computer-Aided Design (CAD) models can be imported and created automatically via a common standard such as Standard for the Exchange of Product Data (STEP) (Wang and Xu 2015), other information including physics, kinematics, sensors, actuators and signal connections, still must be created manually. If a generic data model is developed and 3D emulation models can be fully automatically built in different VC software applications via this software-independent data model, manual remodeling work of 3D emulation models will be significantly reduced in the VC phase, and thus the overall cost of establishing a DT will drastically decrease (Schamp et al. 2018). Under this circumstance, a generic 3D VC-oriented data model is urgently in demand.

AutomationML (AML) is a good way to solve this problem (Lüder, Schmidt, and Rosendahl 2015). AML is short for "Automation Markup Language", which is a standardized data format aimed at integrating multi-disciplinary data and steering data exchange between multiple engineering disciplines (Drath 2021b). It can integrate heterogeneous information into a standard data structure (Drath et al. 2008). To date, geometry, physics and kinematics information can be expressed in the Collada format and integrated into AML (Li et al. 2015; AutomationML 2017b); AML can also be used to model communication-related sensors, actuators and logical devices (AutomationML 2014, 2017a, 2021). However, the current AML standard libraries are not comprehensive enough to achieve full-scope 3D VC-oriented data exchange (Thongnuch and Fay 2017; AutomationML 2023). On one hand, the attributes of 3D VC-related sensors, actuators and signal connections are still not standardized in AML; On the other hand, the linking method between these elements (3D VC-related sensors, actuators and signal connections) and 3D virtual models are unclear. Thus, the current AML standard (AML itself) is not comprehensive enough to realize full-scope interoperability in 3D VC. In this case, the authors suggest extending the current AML standard to make a generic data model to improve the interoperability between different 3D VC software applications. With this extension, VC software developers can develop their software-specific "Import" and "Export" plug-ins, which will significantly reduce the workload of software users when exchanging 3D emulation models between different 3D VC software applications.

The remainder of the paper is structured as follows. In Section 2, the state of the art is illustrated. In accordance with this, it infers that no generic data model exists for the data exchange between different 3D VC software applications, and an extension of the current AML standard is suggested to fill this gap. In Section 3, a case-driven iterative approach is introduced to continuously enrich and improve a generic VC data model, and the attribute naming rules of this data model are also proposed. In Section 4, an initial version of the AML extension is created and its interoperability performance is evaluated. In Section 5, the results are analyzed and discussed. Finally, conclusion and outlook are described in Section 6.

2. State of the art

This section describes the state of the art. In Section 2.1, contemporary industrial standards and neutral data formats for exchanging 3D virtual models are presented. Furthermore, the current research status on the automatic generation of VC models and the realization of 3D emulation model data exchange between different 3D VC software applications is elaborated in Section 2.2.

2.1. *RAMI 4.0 and AML*

In Industry 4.0, heterogeneous multi-disciplinary data are interrelated throughout the product life cycle (Xu, Xu, and Li 2018), and a unified industry-recognized standard data model is required to achieve a coherent and no-data-loss workflow (Nagorny et al. 2020). For this purpose, Reference Architectural Model for Industry 4.0 (RAMI 4.0) was introduced to bridge this gap (Müller et al. 2022). RAMI 4.0 was developed by the German Mechanical Engineering Industry Association (VDMA), the German Electrical and Electronic Manufacturers' Association (ZVEI) and the German Digital Association (Bitkom) (Yli-Ojanperä et al. 2019). It came into the public in 2015 (Hankel and Rexroth 2015). The aim of RAMI 4.0 is to link all crucial data of Industry 4.0 by a 3D architecture, and all the participants in Industry 4.0 can understand each other by speaking the same RAMI 4.0 language (Schweichhart 2016). As is shown in Figure 1, the architecture of RAMI 4.0 consists of three axes (Xu et al. 2021): the "Hierarchy Levels" axis, the "Life Cycle & Value Stream" axis and the "Layers" axis. The "Hierarchy Levels" axis, namely "horizontal integration", describes the connections between resources and facilities within factories. The "Life Cycle & Value Stream" axis, namely "end-to-end integration", expresses the life cycle of a product from conceptual design to finished product. The "Layers" axis, namely "vertical integration", is divided into six different layers, which shows how data are transferred between the physical layer and the digital layer. The "Information" layer is to define all the information of devices in the manufacturing process throughout the product life cycle, and AML is a standard neutral data format recommended to model and exchange data in RAMI 4.0 (Adolphs et al. 2015; Beisheim et al. 2020; Fraile et al. 2019; Drath et al. 2023).

AML was initialized by Daimler AG in 2006 and published at the Hannover fair in 2008 (Lüder et al. 2014). AML aims to solve the "heterogeneous tool landscape" among multiple different engineering disciplines throughout the product life cycle (Drath et al. 2008) and it is regarded as the leading data exchange format for interoperability between engineering tools (ZVEI 2022). Because of its strength in data modeling and data exchange, AML is obtaining growing attention (Zhao et al. 2021). The data structure of AML is based on Computer Aided Engineering Exchange (CAEX), which is an Extensible Markup Language (XML)-based data format. For CAEX version 2, AML data structure can be divided into four parts: Instance Hierarchy (IH), System Unit Class Library (SUCL), Role Class Library (RCL) and Interface Class Library (ICL). IH is the core structure of AML, made up of numerous Internal Element (IE), which contains the main data contents of the project. Each IE is referred to as a Role Class (RC) in RCL to specify the type of each IE, by adding a role-referenced (RR) node in the sub-level of IE. SUCL is a collection of System Unit Class (SUC), which is used to quickly create the IEs in the structure of IH by dragging and dropping. Interface Class (IC), located in ICL, is referred to by an External Interface (EI) which is a sub-

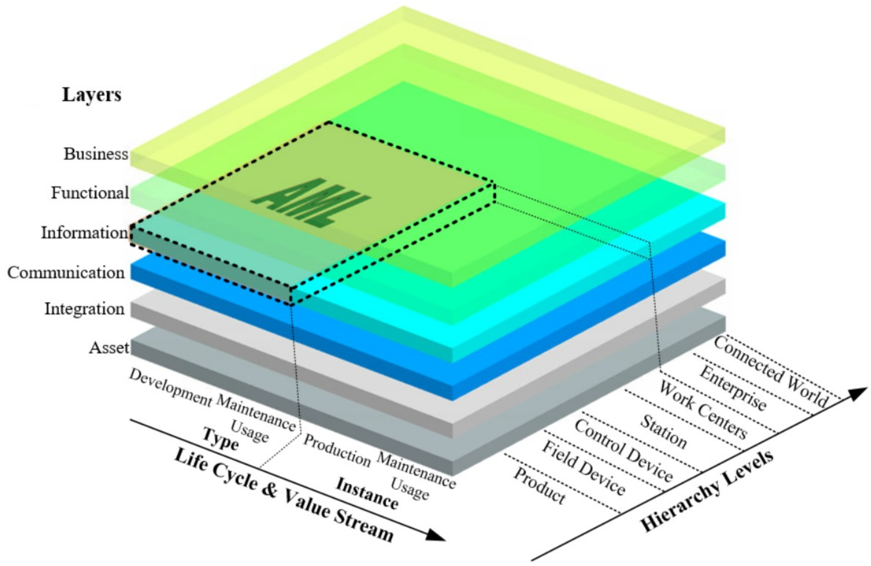


Figure 1. AML in RAMI 4.0 architecture.

node of IE, to link IE with heterogeneous data including geometry and kinematics, logic and behavior, and others (Lüder and Schmidt 2017). Among them, geometry and kinematics information is described in the format of Collada, while logic and behavior information is illustrated in the format of PLCopen XML (Drath 2021a). Collada came into public in 2008, and it is an XML-based interchange format for interactive 3D applications, where kinematics data are included since version 1.5 (Barnes and Finch 2008). PLCopen XML was developed by PLCopen, and it represents logic and behavior information based on XML format (Estévez et al. 2010).

Notably, all AML-related data formats are standardized, of which CAEX, Collada and PLCopen XML belong to IEC 62424, ISO/PAS 17506 and IEC 61131-10 respectively (Holm et al. 2012; Berardinelli et al. 2015; An et al. 2020). AML also belongs to an IEC standard, which is IEC 62714 (Henßen and Schleipen 2014). To date, there are five sub-standards in IEC 62714: Architecture and general requirements of AML, Semantics libraries of AML, Geometry and kinematics of AML, Logic of AML and Communication of AML. Therefore, it can be concluded that AML is a standardized data format to model data and do data exchange.

Although AML is a standardized data format to integrate geometry and kinematics information, however, 3D VC-related sensors, actuators and signal connections are still not comprehensively included in the current AML standard. Besides, as AML is standardized since 2015 (IEC 62714-2, Edition 1.0), it has not been widely utilized to exchange data by most 3D commercial software (Babcinschi et al. 2019). In addition to AML, there are various kinds of neutral formats for 3D virtual model data exchange. Nevertheless, these formats are not comprehensive enough to exchange 3D emulation models between different 3D VC software applications (Zhao, Aghezzaf, and Cottyn 2023).

2.2. *Related work on VC model generation and data exchange*

At this moment, VC can be done in two different ways: Dynamic model-based VC (also known as "1D VC") and 3D VC (Hoffmann et al. 2010; Jackson 2020). 1D VC can test the system based on the real-time values in the behavior models, while 3D VC can verify the feasibility of the automation system based on the movement of 3D geometrical models (Süß, Strahilov, and Diedrich 2015). According to this, VC model generation is divided into two directions: (1) Automatic generation of dynamic models, (2) Automatic generation of 3D emulation models.

For 1D VC, the Functional Mock-up Interface (FMI) is widely used for exchanging dynamic models between different dynamic simulation software tools, such as Simulink and Dymola (FMI 2023; Gunnarsson 2016; Graeser et al. 2011). However, the FMI is mainly aimed at the data exchange of dynamic models, instead of 3D emulation models (Jackson 2019).

For 3D VC, various software applications have been developed in their own cores (Adnan, Daud, and Saud 2014). Under this circumstance, neutral data formats for the interoperability of different 3D software applications were demanded. Since 1960, various 3D neutral data exchange formats have been developed, but they only target on the data exchange of geometry information (Pratt 2001). With the development of 3D simulation software, neutral formats have been developed and improved, and kinematics information has been integrated (Wardhani and Xu 2016). Since 2010, 3D VC software has been introduced (Drath, Weber, and Mauser 2008). A comprehensive 3D emulation model contains not only geometry and kinematics information but also sensors, actuators and signal connections information (Ayani, Ganebäck, and Ng 2018). In this case, it requires more information integrated into the neutral formats. In addition, the standardization of neutral formats is also a key factor. If a neutral format is standardized, it is reliable and can be used directly. According to this, five criteria are chosen to categorize these works, which are as follows:

/C1/ The VC-related data exchange in the work is based on 3D geometrical models.

/C2/ Besides geometry, kinematics information is also included in the data exchange format.

/C3/ Besides geometry and kinematics information, 3D VC-related sensors, actuators and signal connections are also included.

/C4/ The data exchange format is generic and neutral.

/C5/ The data exchange format is developed according to industrial standards.

To automatically generate and exchange VC models, several studies have been conducted. Based on the criteria above, a table of relevant studies is created to intuitively display the analysis results, as shown in Table 1.

As listed in Table 1, no solutions have been found on realizing full-scope interoperability between different 3D VC software applications (according to the results in Column "/C3/"). Therefore, a generic VC data model is crucial to solving this problem. According to the experiment results (Zhao, Aghezzaf, and Cottyn 2023), AML is considered as the best starting point for 3D emulation model generation in 3D VC software applications. Accordingly, the authors dedicate themselves to extending the existing AML standard and proposing a generic VC data model for achieving 3D VC-related interoperability, and this data model is named "AML-VC extension". The location of the AML-VC extension in RAMI 4.0 architecture is shown in Figure 2.

As depicted in Figure 2, the AML-VC extension is located throughout the "Information" layer, "Communication" layer and "Integration" layer of RAMI 4.0 architecture. The "Information" layer contains geometry and kinematics information, the "Communication" layer describes signal connection information, and the "Integration" layer includes the information of sensors and controllers. Besides, the AML-VC extension covers the whole "Life Cycle & Value Stream" axis. It means that the AML-VC extension not only can automatically create 3D emulation models in 3D VC software

Table 1. An analysis of the studies on VC model generation and data exchange.

Authors (Publish Year)	Contributions	/C1/	/C2/	/C3/	/C4/	/C5/
Westkämper et al. (2012)	Automatic generation of a process simulation model based on an Electronic Computer-Aided Design (ECAD) model	N	/	/	/	/
Barth and Fay (2013)	Automatic generation of simulation models based on Piping and Instrumentation Diagrams (P&ID) and CAEX	N	/	/	/	/
Oppelt et al. (2014)	Automatic generation of a simulation model from plant engineering data based on P&ID and XML	N	/	/	/	/
Hoernicke, Fay, and Barth (2015)	Generating simulation models for brown-field projects based on Human-Machine Interface (HMI) graphics and CAEX	N	/	/	/	/
Arroyo et al. (2016)	Automatic generation of qualitative plant simulation models based on P&ID and AML	N	/	/	/	/
Prat et al. (2017)	An automated generation flow of simulation models for checking control/monitoring system based on P&ID and XML	N	/	/	/	/
Martinez et al. (2018)	Automatic generation of a high-fidelity dynamic thermal-hydraulic process simulation model based on an Mechanical CAD (MCAD) plant model and Comma Separated Value (CSV)	N	/	/	/	/
Park et al. (2009)	Automatic generation of a plant simulation model from the symbol table of a Programmable Logic Controller (PLC) program	Y	Y	N	U	Y
Schyja, Bartelt, and Kuhlenkötter (2014)	Data exchange between heterogeneous engineering tools by utilizing AML-based "Smart Components"	Y	Y	N	U	Y
Kiesel et al. (2017)	Automatic generation of VC models based on Graph-Based Design Language (GBDL) and AML	Y	Y	N	U	Y
Zhang, Yan, and Wen (2020)	An information modeling approach for Cyber-physical Production System (CPPS) based on AML	Y	Y	N	U	Y
Beisheim et al. (2021)	Generating DTs of tooling machines by using GBDL and AML	Y	Y	N	U	Y
Kaiser, Reichle, and Verl (2022)	AML-based automatic generation of DT models for reconfigurable manufacturing systems in timber construction	Y	Y	N	U	Y
Li, Tian, and Vogel-Heuser (2019)	AML-based data exchange between SysML4Mechatronics and Creo Parametric	Y	N	N	Y	Y
Breckle et al. (2017)	Automatic generation of 3D layouts based on GBDL	Y	U	N	P	N
Schopper et al. (2021)	Automatic generation of simulation models by using a GBDL-based model called "Executable Integrative Product-Production Model (EIPPM)"	Y	U	N	P	N
Yemenicioglu (2016)	An AML-based framework for the automatic generation of a 3D material handling handling system VC model based on Product-Process-Resource (PPR) concept	Y	Y	N	P	Y
Thongnuch, Fay, and Drath (2018)	Semi-automatic generation of a virtual representation of a production cell based on AML	Y	Y	N	P	Y
Thongnuch (2021)	AML-based generation of kinematics-included VC models by automatically recognizing the component mate information of MCAD	Y	Y	N	Y	Y

Y: Yes, N: No, P: Partially, U: Unclear, /: Not required.

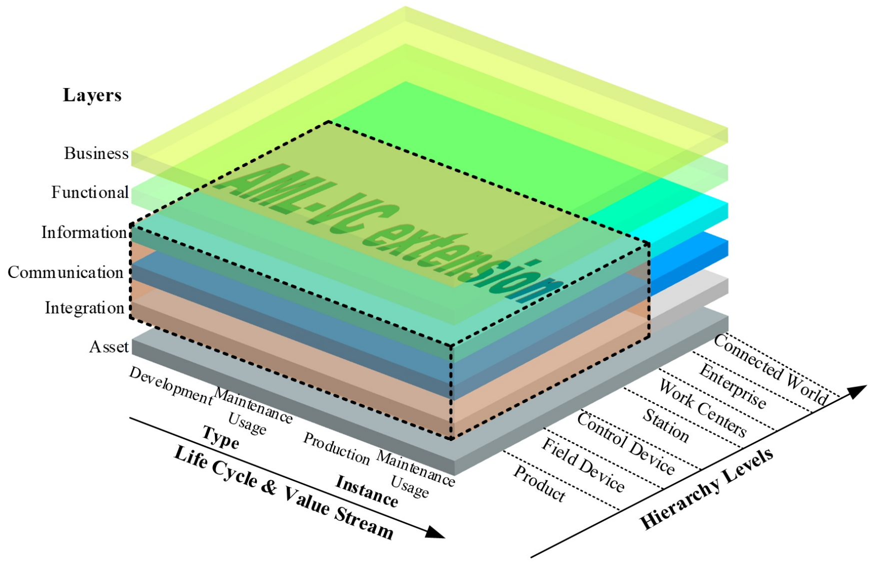


Figure 2. The "AML-VC extension" in RAMI 4.0 architecture.

applications during the product design phase, but also can be generated with real-time data from a 3D VC software application (used as a "digital shadow") during the real manufacturing process. When it comes to the axis of "Hierarchy Levels", the AML-VC extension covers all categories in the engineering field, including "Product", "Field Device", "Control Device", "Station" and "Work Centers". As "Enterprise" and "Connected World" contain more information beyond the scope of engineering, they are not included in the AML-VC extension.

3. Methodology

To develop the AML-VC extension, a methodology is needed. In Section 3.1, a case-driven iterative approach is introduced to continuously evolve towards a comprehensive AML-VC extension. During the iteration, a comparison of attribute names is conducted and attribute naming rules are presented in Section 3.2.

3.1. *A case-driven iterative approach to developing an AML-VC extension*

An AML-VC extension is a generic VC data model, which is a 3D VC-oriented, yet software-independent data structure, containing interrelated information, including geometry, physics, kinematics, sensors, actuators and signal connections. Based on this AML-VC extension, 3D emulation models can be created in a 3D VC software application automatically by developing AML-VC extension-oriented "Import" and "Export" plug-ins with Application Programming Interface (API), regardless of which 3D VC software application is used as the modeling environment.

To create such a comprehensive data model, a case-driven iterative approach is proposed, which is shown in Figure 3. In this approach, the generic VC data model is continuously improved based on new cases. Each case is composed of two aspects: (1) A particular 3D VC software application, which must have never been used previously to develop the generic VC data model; (2) Several application-specific emulation models, among which all the VC modeling-relevant functions of this software application have been used.

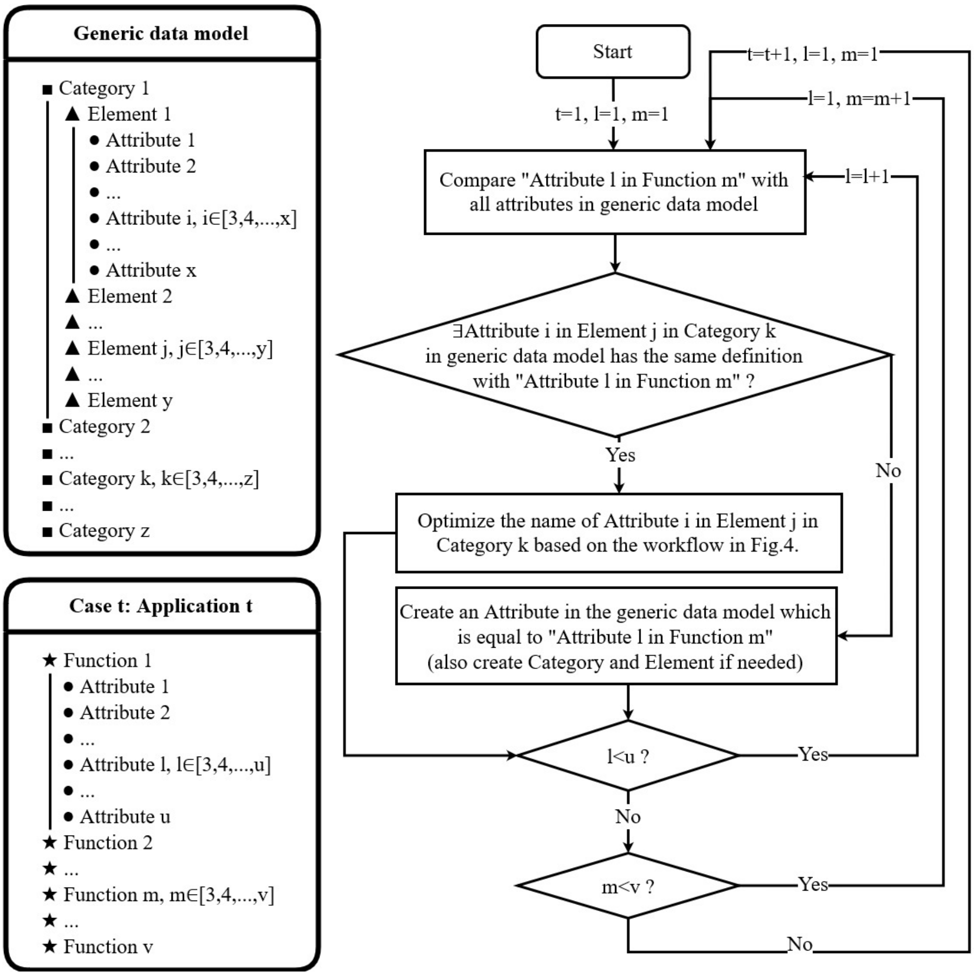


Figure 3. A framework to develop a generic data model in accordance with a case-driven iterative approach.

As depicted in Figure 3, the hierarchy of the generic data model is divided into three levels: “Category”, “Element” and “Attribute”. These categories, elements and attributes are well structured and linked with each other in the generic data model. Based on this, an arbitrary 3D emulation model can be created in a 3D VC software application. Furthermore, 3D VC software applications contribute to developing the generic VC data model. In each case, there is a particular 3D VC software application, containing a variety of functions that are utilized to create software-specific emulation models. Like the hierarchy of the generic data model, there are also various attributes in different functions. These data structures are illustrated in detail with examples in the literature (Zhao, Aghezzaf, and Cottyn 2023).

Besides, a framework for gradually developing a generic VC data model is also described in Figure 3. Based on this framework, the generic data model is continuously evolved by comparing its own attributes with all the VC-modeling function attributes of the 3D VC software application individually. This can be illustrated in a mathematical way. For a generic data model, suppose the quantity of categories is z, and there is a Category k ($k \in [1, z]$), which contains y elements. For one of these elements, Element j ($j \in [1, y]$), there are x attributes attached to it. Similarly, for an arbitrary case t, there is a VC software application t, in which there are v VC-modeling-relevant functions. During the procedure of individual comparison, Function m ($m \in [1, v]$) is one of these functions, and it contains u attributes. If there is an Attribute l ($l \in [1, u]$) in Function m, which has the same definition with an Attribute i ($i \in [1, x]$) in Element j in Category k, then the name of the Attribute i is to be optimized according to the workflow defined in Section 3.2. On the contrary, if no attributes of the generic data model have the same definition with the Attribute l of the application, a new attribute is to be supplemented in the right place of the generic data model, and a new category or element can also be created if necessary. In this way, the generic VC data model is getting increasingly comprehensive as more cases are added.

3.2. *The naming rules of generic VC data model attributes*

As illustrated in Section 3.1, the name of Attribute i in Element j in Category k is required to be optimized when it has the same definition as Attribute l in Function m of a VC software application. The workflow of optimizing a generic VC data model attribute name is described in Figure 4.

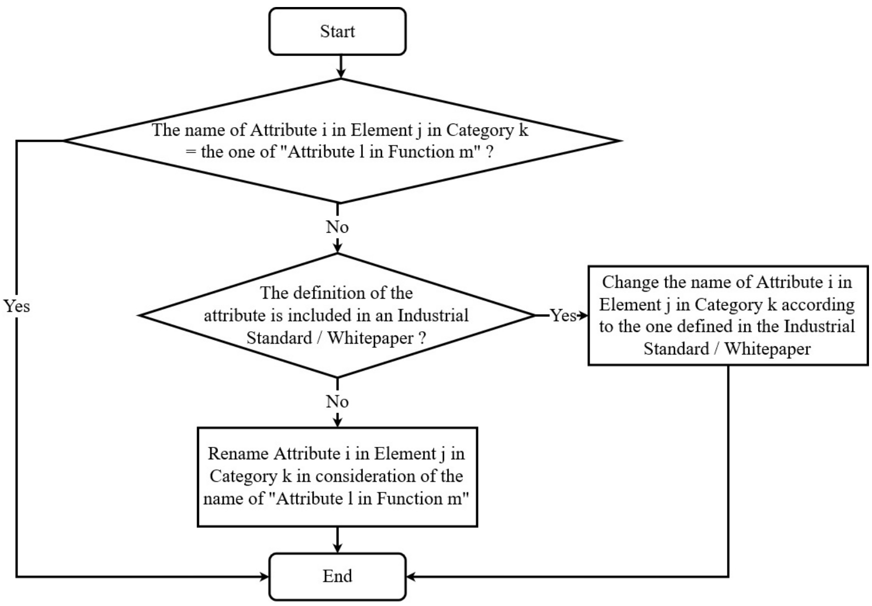


Figure 4. The workflow of optimizing a generic data model attribute name.

The procedure starts with the comparison of both attribute names. If their names are exactly the same, then the name of Attribute i in Element j in Category k is not obliged to change. If not, the name of the Attribute i is to be changed.

Subsequently, if the name of the Attribute i is suggested to change, industrial standards (related whitepapers) are carefully checked. If there is a similar definition of this attribute, then the Attribute i is renamed according to the one defined in industry standards (related whitepapers). For example, there is an attribute called “Sliding” in the generic data model, while there is a so-called “Translational” attribute in a 3D VC software application, both attributes represent the same definition, namely a kind of joint type. To solve this problem, an attribute called “Prismatic” is found in an industrial standard (ISO 17506: 2022). In this case, the relevant generic data model attribute is renamed to “Prismatic” instead of “Sliding” or “Translational”.

However, if no definition of this attribute can be found in industrial standards (related whitepapers), the generic data model attribute is renewed by considering the attribute name located in the 3D VC software application and making a compromise. For instance, there are two similar definition attributes, which are from the generic data model and a 3D VC software application respectively. One attribute is called “Private Key”, while the name of the other attribute is “Use Private Key File”. As a result, “Use Private Key” is used to renew the attribute name in the generic data model.

Based on these naming rules, the attributes of the generic data model are renamed.

4. AML-VC extension

In this section, an initial version of the AML-VC extension is developed, utilized and evaluated. In Section 4.1, a first iteration of the proposed approach is implemented using two 3D VC software applications as cases, namely Siemens NX (SNX) and Visual Components (VCO). Based on this, an initial version of the AML extension is created. Next, an AML-VC extension-based data exchange method between SNX and VCO is described and verified in Section 4.2. The interoperability performance is subsequently evaluated by exchanging 10 3D emulation models between SNX and VCO via this extension in Section 4.3.

4.1. *Creation of an AML-VC extension based on a first iteration*

As proposed in Section 2, an AML-VC extension is considered a suitable data format to build a VC data model and to do data exchange between different 3D VC software applications. Thus, based on the case-driven iterative approach (Section 3.1) and the attribute naming rules (Section 3.2), a preliminary AML-VC extension is created by using AML and two 3D VC software applications, namely SNX and VCO, as cases for this first iteration.

During the iteration phase, an AML-VC extension is gradually developed by taking the union of all the VC-related functions and attributes of SNX and VCO. Among them, 92 VC-related functions and 1350 relevant attributes exist in SNX, while the numbers are 34 and 381 respectively in VCO. After the iteration including attribute name optimization, a first version of the AML-VC extension is created, with 7 categories, 115 elements and 1670 attributes. The data structure of the initial AML-VC extension is briefly shown in Figure 5.

As depicted in Figure 5, a set of libraries has been created, including Role Class Library “AutomationMLVirtualCommissioningRoleClassLib”, Interface Class Library “AutomationMLVirtualCommissioningInterfaceClassLib” and Attribute Type Library “AutomationMLVirtualCommissioningAttributeTypeLib”. In these libraries, a variety of 3D VC-related elements have been defined. Moreover, a large amount of 3D VC-related information is composed in these elements, not only geometry, physics and kinematics information but also information of sensors, actuators and signal connections. Furthermore, the elements in these libraries are linked with each other, which makes the AML-VC extension relatively comprehensive to express 3D emulation models.

As 3D VC-related sensors, actuators and signal connections are additional elements in this AML-VC extension, how to integrate these kinds of information on the IE and

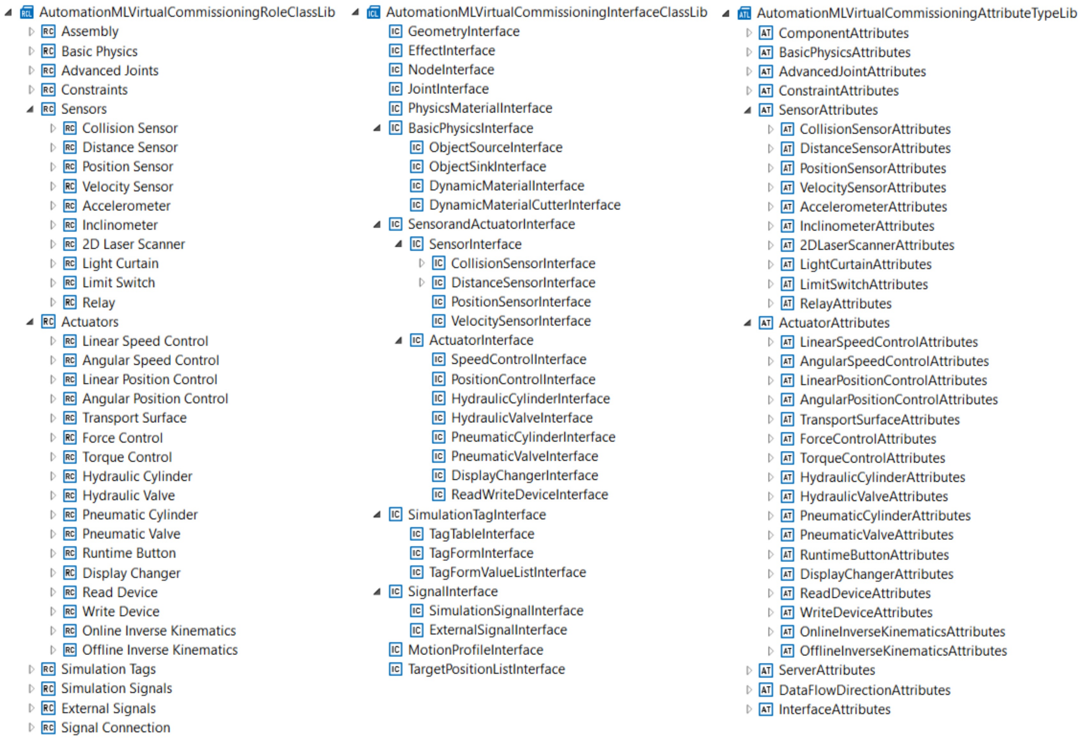


Figure 5. An AML-VC extension created by the iteration of two cases, SNX and VCO.

attach them to IH is also crucial.

In terms of 3D VC-related sensor information, three kinds of sensors are taken as an example, which are "Distance Sensor", "Position Sensor" and "Velocity Sensor". The information of the sensors is structured into AML instead of Collada. One reason is that there has not been an update in Collada since 2008, while the other reason is that the additional information can be a new RC in RCL, which provides great convenience for quickly building AML-based data models. Figure 6 shows the data structure of "Distance Sensor", "Position Sensor" and "Velocity Sensor" in the AML-VC extension. Based on the principle of taking union described in Section 3.2, the attributes of these three kinds of sensors in both SNX and VCO are combined and structured into AML. As Figure 6 shows, each sensor is a sub-IE under its parent IE "Sensors". The sub-IE "Distance sensor" refers to a "Raycast Sensor" in VC, while it is a "Distance Sensor" in SNX. The distance sensor can be attached to a certain assembly link, which can be referred to as a "node" in Collada, and real-time distance value and trigger value can be linked to relevant simulation signals via Internal Link (IL)s. The start point and the direction of the distance sensor are defined by the attribute "Frame", in which "x", "y", and "z" of the frame are the start point and the positive Z-axis of the frame in the direction. The other attributes of the "Distance Sensor" in an AML-VC extension are taken from both SNX and VCO. Besides, position sensors and velocity sensors can indicate the real-time position and speed of joints. Thus, EI is used to refer to the relevant joint in Collada. At the same time, IL is used to connect the sensor to a simulation signal. It is worth noting that no concrete definitions of a position sensor or a velocity sensor in VCO can be found while there are relevant definition modules in SNX. Therefore, all attributes of "Position Sensor" and "Velocity Sensor" are defined according to the definitions of the relevant modules in SNX.

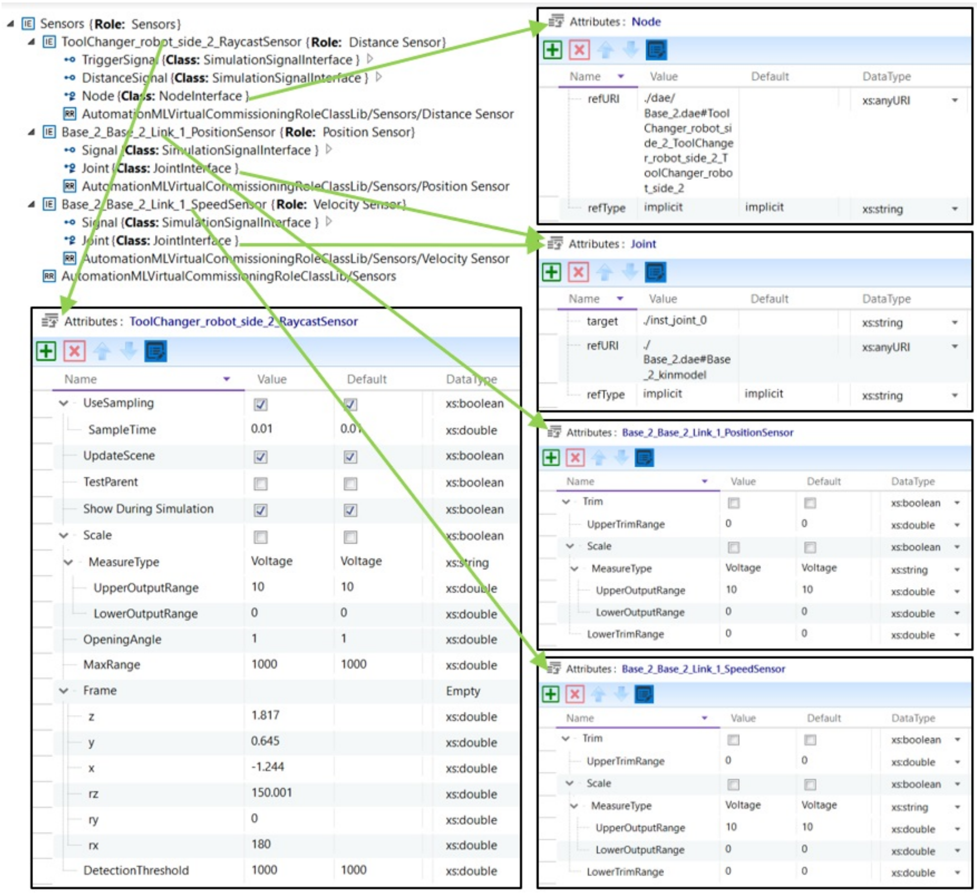


Figure 6. 3D VC-related distance sensor, position sensor and velocity sensor-related definitions in the AML-VC extension.

In terms of 3D VC-related actuators, like the definition of sensors, each actuator is a sub-IE under the parent IE "Actuators". As shown in Figure 7, "Speed Control" is taken as an example of actuators. As each actuator is attached to a joint, an EI is added to the IE of "Speed Control", and the EI is referred to the relevant joint in Collada. An IL can also be added to connect the "Speed Control" with an input signal of the simulation model so that the value of "Speed Control" can be changed in real time according to the input signal value. The attributes of "Speed Control" are stored in both AML and Collada. The reason is that some attributes of "Speed Control" are already a part of the standard data structure under the "motion" of "library_articulated_systems" in Collada, and these attributes are regarded as motion attributes of kinematic joints. Therefore, the new attributes of "Speed Control" from SNX and VCO are added to "newparam" nodes of relevant joint axes in Collada, such as "forward limit", "reverse limit", "lag time" and "settle time". However, the execution speed of "Speed Control" does not belong to a joint attribute, so it is added to an attribute of "Speed Control" IE in AML.

Besides sensors and actuators, 3D VC-related signal connections also play an important role. For signal connections, OPC United Architecture (OPC UA) communication protocol is taken as an example. As is shown in Figure 8, sensors and actuators on this model are connected to an OPC server via the OPC UA communication protocol. At the same time, the sensors and actuators on this model are mapped with different

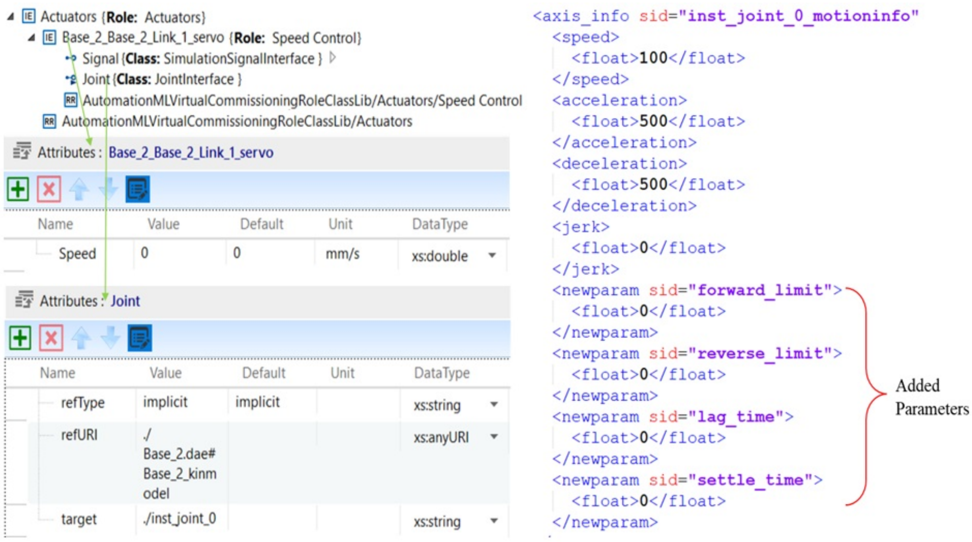


Figure 7. 3D VC-related speed control definitions in the AML-VC extension.

tags in the Open Platform Communications (OPC) server by signal mapping. In this case, the real-time values of the sensors can be read from the tags on the OPC server, while the speed controls on this model can also be driven in real time according to the values of the tags on the OPC server. As signal connections are not a part of kinematics, the information of signal connections is recommended to be structured into AML instead of Collada. Like the hierarchies of "Sensors" and "Actuators" in AML, the IE of "OPC UA" is a sub-IE of "Connectivity" IE. Furthermore, the IEs of "OPCUAserver" are sub-IEs of the "OPC UA" IE as it is possible for several OPC servers to connect to the VC model via OPC UA. Based on the signal data flow, An IE of "Simulation to server" and an IE of "Server to simulation" are located under the IE "OPCUAserver". The signal data flow of "Simulation to server" means the signal values on the simulation model are sent to the tags on the OPC server, while the signal data flow of "Server to simulation" means the values of the tags on the OPC server are sent to the simulation model. Under each data flow IE, the IE of "Simulation Signals" and the IE of "External Signals" are attached respectively. Both have sub-IEs of "Simulation Signal"s and "External Signal"s, and ILs are used to pair the relevant signals respectively. Based on this, the signals on the sensors belong to the sub-IEs under the "Simulation to server" IE, while the signals on the actuators belong to the "Server to simulation" IE. Moreover, the attributes of the "OPCUAserver" IE and the two signal data flows are structured based on the union method between SNX and VCO.

Based on the definitions of 3D VC-related sensors, actuators and signal connections, a general data structure of an AML-VC extension is created. The IH is the root node of the AML-VC extension, which is the collection of all the data in the 3D VC model. The data connection IH contains four different kinds of IEs, namely "Component", "Sensors", "Actuators" and "Connectivity". The IE of "Component" represents the assembly hierarchy of the 3D VC model, and it can contain several sub-IEs. All these IEs can refer to different Collada files via EI, and the geometry and kinematics information is stored in Collada. As it is illustrated above, the IEs of "Sensors", "Actuators" and "Connectivity" contain the additional necessary information in the 3D VC model, namely the information of sensors, actuators and signal connections. Based on this, a

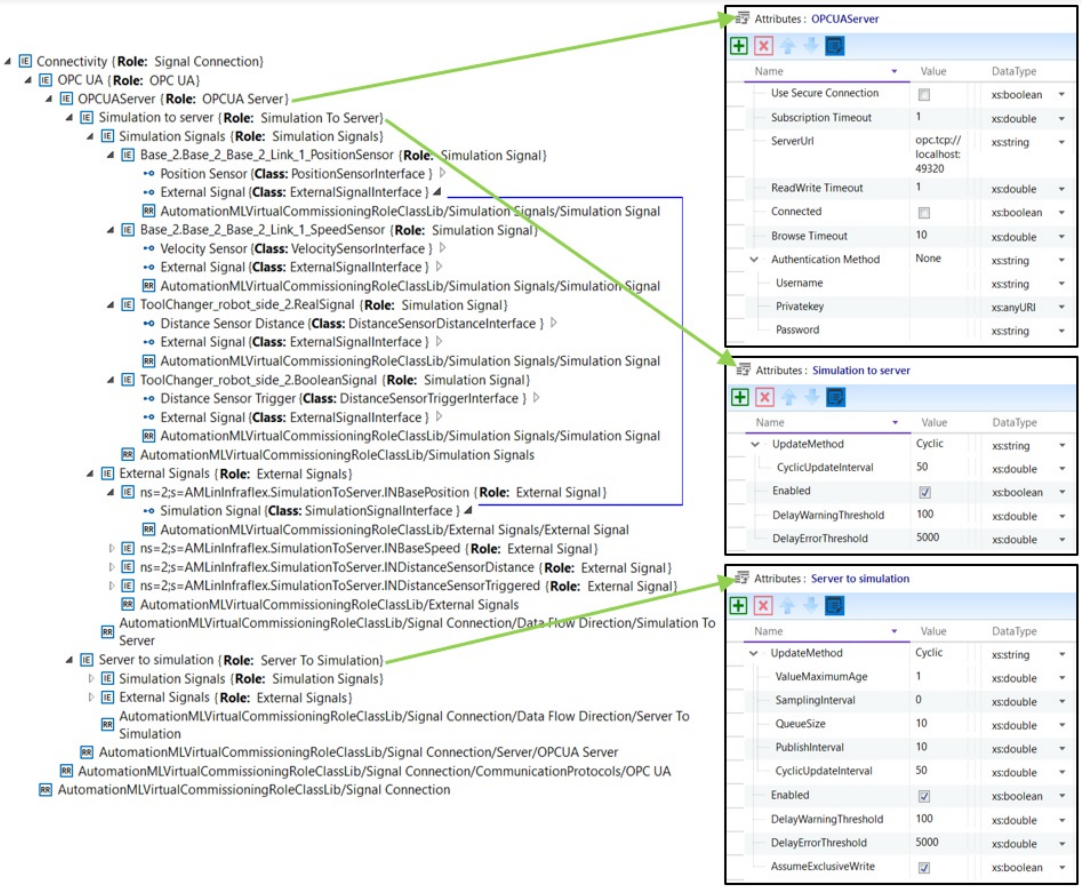


Figure 8. 3D VC-related OPC UA signal connection definitions in the AML-VC extension.

first version of the AML-VC extension is created.

4.2. *AML-VC extension-based data exchange between SNX and VCO*

A comprehensive AML-VC extension contributes to representing 3D emulation models. Nevertheless, due to the commercial competition among different 3D VC software companies, each 3D VC software company develops its own 3D VC model format, and the specific 3D VC model format can only be loaded and displayed by the corresponding 3D VC software application. Under this circumstance, an AML-VC extension can neither be successfully evaluated nor be widely promoted if there are difficulties in generating it from VC software applications. To solve this problem, a method for realizing data exchange between different VC software applications based on AML-VC extension is required.

In this case, two 3D VC software applications, SNX and VCO, are chosen to develop the method for exchanging 3D emulation models between different 3D VC software applications. Both of these software applications are capable to do VC, but they are developed by different software companies and they have different software cores. SNX is integrated with various function modules, which include many different aspects, such as product design, machining design, mechatronics design and process simulation. In these models, the module “Mechatronics Concept Designer (MCD)” is aimed at conducting VC. Contrary to SNX, VCO is mainly focused on quickly building virtual automation systems and conducting manufacturing process simulation. To realize the AML-VC extension-based data exchange between SNX and VCO, a method has been developed, which is shown in Figure 9.

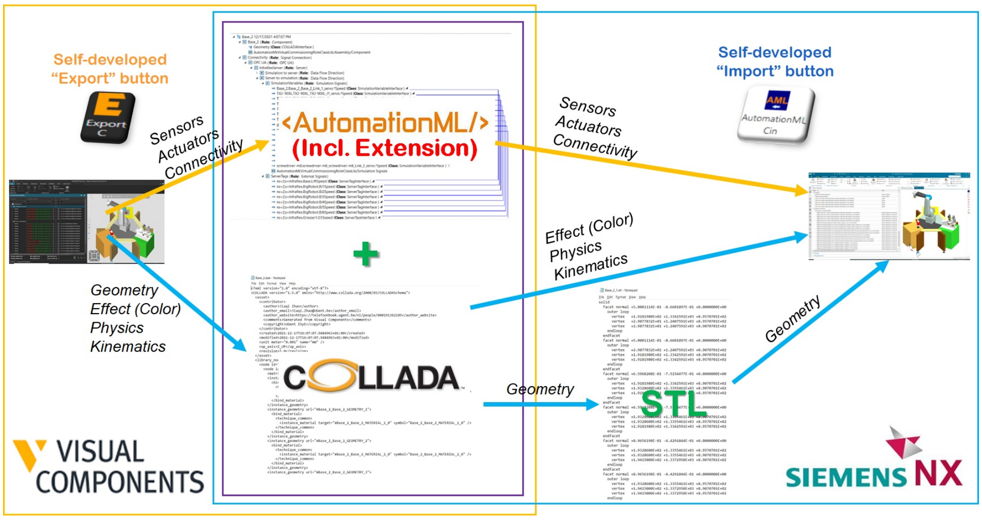
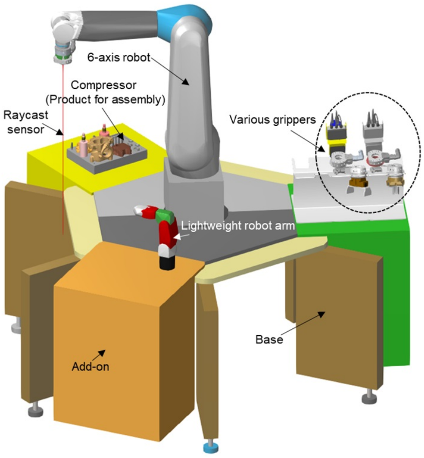


Figure 9. A method for AML-VC extension-based data transmission from VCO to SNX.

As depicted in Figure 9, a method for AML-VC extension-based data transmission from VCO to SNX is described. As the AML format is not recognized by these two kinds of software, "Import" and "Export" plug-ins are developed in both software applications respectively with their software-specific APIs according to the AML-VC extension data structure. In this method, an AML-VC extension is generated from VCO by clicking the self-defined "Export" button. In this model, geometry, color, physics, and kinematics information is stored in Collada files, while the sensors, actuators and signal connection information is contained in the AML file. The Collada files are linked to the AML file via EIs. Then, the AML-VC extension is imported into SNX via the self-developed "Import" button in SNX, and the relevant 3D emulation model is generated in SNX. During the import, geometry information is firstly converted into STL format, then loaded into SNX, as the import of STL files can save a lot of time in comparison with creating the mesh points by coding. Similarly, the AML-VC extension-based data transmission from SNX to VCO is exactly an inversion of this method.

To verify the applicability of the AML-VC extension-based data exchange method, a 3D emulation model of a "Flexible Assembly Work Cell (FAWC)" is taken as an example to do data exchange between SNX and VCO. The VC-related descriptions of this emulation model are shown in Figure 10. In this model, a distance sensor, a position sensor and a velocity sensor are attached to detect the real-time data on the work cell, while several actuators are also deployed on it to control execution speeds of corresponding prismatic joints and revolute joints. Input and output signals are also defined in this model, and the signals are mapped with the tags in the OPC server via OPC UA communication protocol. Based on this, real-time parameters of the sensors can be observed in the OPC server while execution speeds of the actuators in the VC model can also be in real-time control by changing the values of tags in the OPC server.

The FAWC-based verification procedure is executed according to the data exchange



6-axis robot		
Category	ID	Type
Joint with Velocity Actuator	J1	Revolute Joint
	J2	Revolute Joint
	J3	Revolute Joint
	J4	Revolute Joint
	J5	Revolute Joint
	J6	Revolute Joint
Distance Sensor	TS	Trigger Signal (Out)
	DS	Distance Signal (Out)

Lightweight robot arm		
Category	ID	Type
Joint with Velocity Actuator	A1	Revolute Joint
	A2	Revolute Joint
	A3	Revolute Joint
	A4	Revolute Joint
	A5	Revolute Joint
	A6	Revolute Joint

Base		
Category	ID	Type
Joint	L1	Prismatic Joint
Velocity Actuator	VB	Input Signal
Speed Sensor	SS	Output Signal

Gripper for clamp		
Category	ID	Type
Inter-related Joint	C1	Prismatic Joint
	C2	Prismatic Joint
	(C3)	(Prismatic Joint)
Velocity Actuator	VC	Input Signal

Gripper for screw		
Category	ID	Type
Joint	S1	Prismatic Joint
	S2	Revolute Joint
Velocity Actuator	V1	Input Signal
	V2	Input Signal

Figure 10. The FAWC model and its VC-related definitions.

framework illustrated in Figure 9. Firstly, a FAWC emulation model is manually created in VCO. Secondly, the AML-VC extension of FAWC is generated by clicking the "Export" button in VCO. Thirdly, the AML-VC extension of FAWC is imported into SNX by clicking the "Import" button in SNX. Finally, a new 3D emulation model of FAWC including exactly the same functions and attributes is automatically created in SNX. After this, real-time verification has been conducted. Figure 11 shows the real-time running comparison of the two FAWC emulation models in SNX and VCO during emulation. The newly generated FAWC model in SNX is moving in the same way as the original FAWC model in VCO, and the movements of both models are driven by the tag values in the OPC server (KEPServerEX). The success of this verification indicates that the method for AML-VC extension-based data exchange between SNX and VCO is feasible.[1]

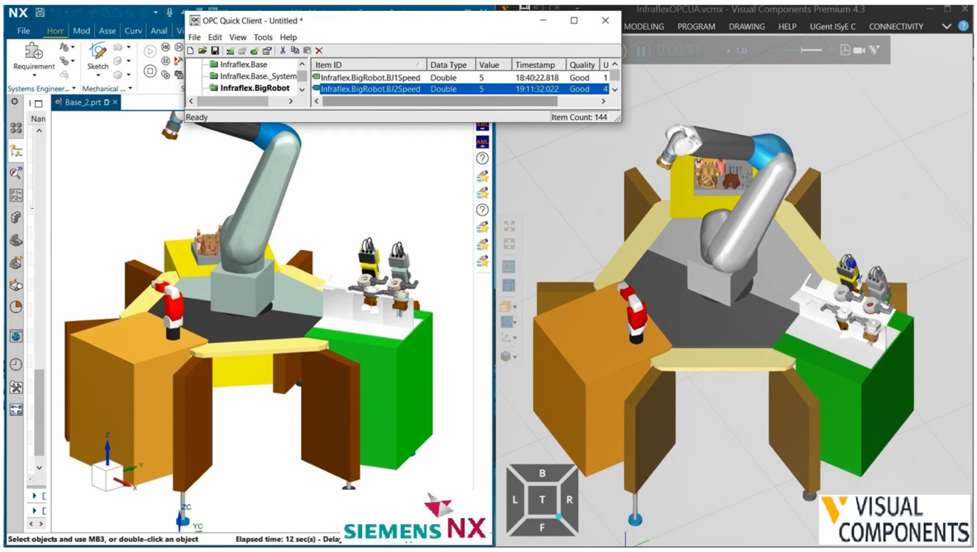


Figure 11. Real-time running comparison of the FAWC model in SNX and VCO.

[1] A recorded video of the execution process is accessible at: https://www.youtube.com/watch?v=PRJ1umTEKqo

4.3. *Interoperability performance evaluation of the AML-VC extension based on two cases*

To evaluate the interoperability performance of the AML-VC extension, two 3D VC software applications, SNX and VCO, are employed as cases. In each case, five 3D emulation models covering the most diverse VC-related functions are selected from the modeling libraries of SNX and VCO respectively (Zhao, Aghezzaf, and Cottyn 2023). Based on this, an interoperability performance evaluation is subsequently conducted by exchanging the 10 selected 3D emulation models between SNX and VCO via the AML-VC extension. With the self-developed "Import" and "Export" plug-ins in SNX and VCO, AML-VC extension-based data exchange between SNX and VCO is achieved. According to the case-based evaluation framework described in the literature (Zhao, Aghezzaf, and Cottyn 2023), the interoperability performance of the AML-VC extension can be evaluated by the attribute conversion rates of the emulation models during AML-VC extension-based data exchange between SNX and VCO. The attribute conversion rates of emulation models emerge during the "Import" and "Export" interaction between the AML-VC extension and a 3D VC software application. The interoperability performance between the AML-VC extension and a 3D VC software application is positively correlated with the average attribute conversion rates of the emulation models during data exchange.

After conducting the evaluation experiment, the attribute quantities and conversion rates of five emulation models during AML-VC extension-based data exchange from SNX to VCO are listed in Table 2, while those of the other five emulation models during AML-VC extension-based data exchange from VCO to SNX are listed in Table 3. In order to present an intuitive comparison, the attribute quantities and conversion rates of the same 10 emulation models during data exchange between SNX and VCO via AML (without extension) are also presented in Table 2 and Table 3 respectively.

Table 2. Attribute quantities and conversion rates of five SNX emulation models during data exchange from SNX to VCO based on AML and AML-VC extension.

Emulation model number	1	2	3	4	5
Attribute quantity in SNX	557	1331	737	984	2684
Attribute quantity in AML	151	459	317	351	1094
Attribute quantity in AML-VC extension	544	1306	732	964	2589
Attribute quantity in VCO converted via AML	126	389	282	235	823
Attribute quantity in VCO converted via AML-VC extension	317	771	399	500	1609
Conversion rate from SNX to AML (%)	27.1	34.5	43.0	35.7	40.8
Conversion rate from SNX to AML-VC extension (%)	97.7	98.1	99.3	98.0	96.5
Conversion rate from SNX to VCO via AML (%)	22.6	29.2	38.3	23.9	30.7
Conversion rate from SNX to VCO via AML-VC extension (%)	56.9	57.9	54.1	50.8	60.0

For AML-VC extension-based data exchange, as shown in Table 2, the average attribute conversion rate from SNX to AML-VC extension is 97.9%, while that from SNX to VCO is 56.0%. Similarly, as shown in Table 3, the percentage from VCO to

Table 3. Attribute quantities and conversion rates of five VCO emulation models during data exchange from VCO to SNX based on AML and AML-VC extension.

Emulation model number	6	7	8	9	10
Attribute quantity in VCO	792	410	526	629	235
Attribute quantity in AML	158	81	246	78	0
Attribute quantity in AML-VC extension	775	406	485	612	229
Attribute quantity in SNX converted via AML	142	80	228	72	0
Attribute quantity in SNX converted via AML-VC extension	378	142	271	280	101
Conversion rate from VCO to AML (%)	20.0	19.8	46.8	12.4	0.0
Conversion rate from VCO to AML-VC extension (%)	97.9	99.0	92.2	97.3	97.5
Conversion rate from VCO to SNX via AML (%)	17.9	19.5	43.4	11.5	0.0
Conversion rate from VCO to SNX via AML-VC extension (%)	47.7	34.6	51.5	44.5	43.0

AML-VC extension is 97.0%, while that from VCO to SNX is 46.1%. When it comes to AML-based data exchange, as listed in Table 2, the attribute average conversion rate from SNX to AML is 36.2%, while that from SNX to VCO is 28.9%. Likewise, the percentage from VCO to AML is 19.8% and that from VCO to SNX is 18.5%, according to Table 3.

5. Discussion

According to the results in Section 4.3, the average attribute conversion rates from SNX to AML-VC extension and that from VCO to AML-VC extension are over 97%, which are much higher than the values via AML (36.2% and 19.8% respectively). It shows that AML-VC extension-based data exchange makes significant progress in the interoperability between SNX and VCO, compared to the data exchange via AML. As the AML-VC extension is created by taking the union of all functions and attributes in SNX and VCO, the attribute conversion rates of the emulation models during the data exchange between the AML-VC extension and SNX (or VCO) should logically be 100%. However, due to the limitations of the current APIs in SNX and VCO, some VC-related functions and attributes are not accessible. Because of this, these functions and attributes cannot be converted and attached to the structure of the AML-VC extension, which causes a decrease in the attribute conversion rates. Besides, script-based attribute definition is also a reason for low attribute conversion rates. For instance, as for emulation model 8, the attribute conversion rate from VCO to AML-VC extension is 92.2%, which is around 5% lower than the rates of other percentages at the same level. The reason is that more “python script” functions are included in emulation model 8 than in the other VCO emulation models. As the syntax in “python script” function is not structured in a fixed format, it is extremely difficult to locate and identify relevant attribute values. Due to this, these script-based attributes are also missing in the AML-VC extension.

As shown in the results, around half of the attributes can be automatically converted between SNX and VCO by using AML-VC extension and application-specific plug-ins. It demonstrates that the interoperability performance between SNX and VCO is around 50% by means of AML-VC extension, which is more than 25% higher than AML-based data exchange (23.7%). It is clear that AML-VC extension-based data exchange converts more attributes during the data exchange between different 3D VC software applications compared to that via AML. The reason is that AML-VC extension integrates more 3D VC-related information, such as sensors, actuators and signal connections, which is included in the current AML standard. It can also be concluded that the attribute similarity between SNX and VCO is around 50%, as about half of the attributes in SNX (or VCO) can be represented in VCO (or SNX). From another perspective, nearly half of the attributes from one VC application are not applicable to the other application. For example, a "Cylinder" shaped collision body can be defined in SNX, but it is not supported in VCO. On the contrary, an elastomer is possible to be described in VCO, while it is extremely difficult to be expressed in SNX. The reason is that SNX and VCO have their own target customers and they are developed in different software cores. SNX contains more VC-related user-defined function modules for making 3D emulation models, while more script-based attribute definition is used in VCO. Furthermore, VCO is a library-based VC software application, in which a 3D emulation model can be easily created by dragging and dropping sub-models from the emulation model library. However, in SNX, 3D emulation models are usually created by defining VC-related functions based on manually created 3D models. Therefore, as each application has its own target users and specific core, attribute conversion rates vary from different pairs of VC software applications. The attribute conversion rate between two VC software applications is proportional to the attribute similarity between them. As the AML-VC extension is developed by taking the union from each case (3D VC software), it contains as many VC-related functions and attributes as possible. This comprehensive extension contributes to the reuse of data when exchanging 3D emulation models between different 3D VC software. To indicate those non-convertible functions and attributes during the data exchange, log files are generated.

6. Conclusion and outlook

In this paper, a first implementation of AML-VC extension for data exchange between different 3D VC software applications is presented. The AML-VC extension is developed by extending the current AML standard. The AML-VC extension contains not only geometry, physics and kinematics information, but also 3D VC-related additional information, such as sensors, actuators and signal connections. A case-driven iterative approach is proposed to continuously improve the comprehensiveness of the AML-VC extension. In this approach, 3D VC software applications are used as cases to further define the AML-VC extension. Moreover, naming rules for AML-VC extension attributes are also introduced. Based on the case-driven iterative approach and the attribute naming rules, an initial version of the AML-VC extension is developed by taking the unions of functions and attributes in two 3D VC software applications, SNX and VCO. To build the AML-VC extension-based data exchange "bridge" between SNX and VCO, plug-ins are developed in SNX and VCO respectively. Due to the limitations of current software-specific APIs and script-based attribute definition, about 3% of 3D VC-related attributes in SNX and VCO cannot be extracted into the AML-VC extension by the plug-ins. The interoperability performance of the

AML-VC extension is subsequently evaluated by exchanging 10 3D emulation models between SNX and VCO via the plug-ins. The experiment results show that AML-VC extension-based data exchange converts an average of nearly 70% more attributes than that based on AML (without extension). It can be concluded that an AML-VC extension makes significant contributions to exchanging 3D emulation models between different 3D VC software applications.

The AML-VC extension-based data exchange creates a new way for interoperability between different 3D VC software applications. The traditional method is to convert a 3D emulation model into STEP format and import it into another 3D VC software application. With this method, all VC-related non-geometry information will be lost. An AML-VC extension can solve this problem by keeping more VC-related information during the data exchange. Besides, the AML-VC extension is completely open source, as it is a combination of AML and Collada formats, which are both XML-based and machine-readable. In this case, the AML-VC extension can be easily read and edited via a text-oriented editor instead of a 3D VC software application. Furthermore, the AML-VC extension can be referenced by 3D VC software companies to develop AML-VC extension-based data exchange plug-ins for their software applications. Based on the plug-ins, 3D emulation models can be automatically generated in relevant 3D VC software by "one button play", which greatly improves efficiency and saves time. In this paper, the AML-VC extension-based data exchange plug-ins have been successfully developed in SNX and VCO, which gives benefits for the AML-VC extension-based data exchange between both software applications. In addition, a case-driven iterative approach has been proposed to continuously improve the comprehensiveness of the AML-VC extension.

Nevertheless, some challenges and uncertainties still exist in the development of the AML-VC extension. Firstly, as an important part of the AML-VC extension, the Collada format has not been updated since 2008. Though kinematics information can be integrated into Collada, it is really difficult to find relevant software that can preview Collada kinematics. This creates a great impediment to the development of the AML-VC extension. Secondly, few 3D VC software applications contemporarily support AML-oriented interfaces. In this case, to realize AML-VC extension-based interoperability between different 3D VC software applications, developing software-specific plug-ins is a must. As different kinds of 3D VC software are developed by various software companies, the software architectures and development languages of these VC software applications also vary, which is extremely cumbersome for "bridge" builders to develop all kinds of "Import" and "Export" interfaces. The required effort in developing such interfaces varies from person to person, depending on the familiarity with the relevant 3D VC software, AML, Collada and programming skills. Thirdly, to create plug-ins, the openness of the software's API library is very critical. If the API library of the software does not support enough development functions, the corresponding plug-in cannot be successfully developed. Additionally, plug-in development also consumes a lot of time in learning the relevant development methods and API functions. Fourthly, it takes effort to improve the comprehensiveness of the AML-VC extension during each case by manually checking standards. However, it is even challenging to conduct it in an automatic way, as synonyms are difficult to be automatically distinguished. Fifthly, so far there is no rule-based checking mechanism for the AML-VC extension. For example, the names of the assembly nodes in VCO must be different from each other, which cannot be detected at this moment.

In future work, a third 3D VC software application will be introduced to evaluate the comprehensiveness of the current AML-VC extension by comparing attribute differences between the 3D VC software and the AML-VC extension. After this, more software applications with their software-specific emulation models will be utilized to continuously improve the AML-VC extension based on the case-driven iterative approach. As the attribute naming rules in the case-driven iterative approach are manual and error-prone, a knowledge-based automatic attribute naming strategy will be developed to automatically give naming cues based on existing data relationships. Besides, a rule-based checking mechanism for the AML-VC extension is to be developed. The ultimate goal of the authors is to make this extension part of a common industry standard, which will require some more formal process.

Disclosure statement

The authors have no relevant financial or non-financial interests to disclose.

Funding

This work was funded by the China Scholarship Council (CSC), which is a non-profit institution affiliated with the Ministry of Education of China.

References

Adnan, Mohd Fahmi, Mohd Fadzil Daud, and Muhammad Sukri Saud. 2014. “Contextual Knowledge in Three Dimensional Computer Aided Design (3D CAD) Modeling: A Literature Review and Conceptual Framework.” In *2014 International Conference on Teaching and Learning in Computing and Engineering*, apr. IEEE. https://doi.org/10.1109/latice.2014.41.

Adolphs, Peter, Heinz Bedenbender, Dagmar Dirzus, M Ehlich, U Epple, M Hankel, R Heidel, et al. 2015. “Status report-reference architecture model industrie 4.0 (rami4.0).” *VDI-Verein Deutscher Ingenieure eV and ZVEI-German Electrical and Electronic Manufacturers Association, Tech. Rep* https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report__Reference_Archtitecture_Model_Industrie_4.0__RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf.

An, Yameng, Feiwei Qin, Baiping Chen, Rene Simon, and Huifeng Wu. 2020. “OntoPLC: semantic model of PLC programs for code exchange and software reuse.” *IEEE Transactions on Industrial Informatics* 17 (3): 1702–1711. https://doi.org/10.1109/TII.2020.2997360.

Arroyo, Esteban, Mario Hoernicke, Pablo Rodríguez, and Alexander Fay. 2016. “Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams.” *Computers & Chemical Engineering* 92: 112–132. https://doi.org/10.1016/j.compchemeng.2016.04.040.

AutomationML. 2014. *Whitepaper Part 5 - Communication.* Technical Report. AutomationML consortium. https://www.automationml.org/wp-content/uploads/2021/06/WP_Communication_V1.0.0.zip.

AutomationML. 2017a. *Best Practice Recommendation: DataVariable.* Technical Report. AutomationML consortium. https://www.automationml.org/wp-content/uploads/2022/05/BPR_007E_BPR_DataVariable_V1.0.0.zip.

AutomationML. 2017b. *Whitepaper Part 3 - Geometry and Kinematics.* Technical Report. AutomationML consortium. https://www.automationml.org/wp-content/uploads/2021/06/AML_Whitepaper_GeometryKinematics_V2.0.0.zip.

AutomationML. 2021. *Application Recommendations: Automation Project Configuration.* Technical Report. AutomationML consortium. https://www.automationml.org/wp-content/uploads/2021/11/AR-APC-1_3_0.zip.

AutomationML. 2023. *Application Recommendation: Toolchain.* Technical Report. AutomationML consortium. https://www.automationml.org/wp-content/uploads/2023/06/AR_Toolchain.pdf.

Ayani, M., M. Ganebäck, and Amos H.C. Ng. 2018. "Digital Twin: Applying emulation for machine reconditioning." In *51st CIRP Conference on Manufacturing Systems*, Vol. 72, 243–248. https://doi.org/10.1016/j.procir.2018.03.139.

Babcinschi, Mihail, Bernardo Freire, Pedro Neto, Lucía Alonso Ferreira, Baltasar Lodeiro Señaris, and Félix Vidal. 2019. "AutomationML for Data Exchange in the Robotic Process of Metal Additive Manufacturing." In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 65–70. IEEE. https://doi.org/10.1109/ETFA.2019.8869079.

Barbieri, Giacomo, Alberto Bertuzzi, Andrea Capriotti, Lorenzo Ragazzini, David Gutierrez, Elisa Negri, and Luca Fumagalli. 2021. "A virtual commissioning based methodology to integrate digital twins into manufacturing systems." *Production Engineering* 15 (3): 397–412. https://doi.org/10.1007/s11740-021-01037-3.

Barnes, Mark, and Ellen Levy Finch. 2008. "COLLADA–digital asset schema release 1.5.0." *Specification, Khronos Group* https://www.khronos.org/files/collada_spec_1_5.pdf.

Barth, Mike, and Alexander Fay. 2013. "Automated generation of simulation models for control code tests." *Control Engineering Practice* 21 (2): 218–230. https://doi.org/10.1016/j.conengprac.2012.09.022.

Beisheim, Nicolai, Markus Kiesel, Markus Linde, and Tobias Ott. 2020. "Using AutomationML and Graph-Based Design Languages for Automatic Generation of Digital Twins of Cyber-Physical Systems." In *Transdisciplinary Engineering for Complex Socio-technical Systems–Real-life Applications*, 135–142. IOS Press. https://doi.org/10.3233/ATDE200070.

Beisheim, Nicolai, Markus Linde, Tobias Ott, and Sebastian Amann. 2021. "Using AutomationML to Generate Digital Twins of Tooling Machines for the Purpose of Developing Energy Efficient Production Systems." In *Transdisciplinary Engineering for Resilience: Responding to System Disruptions*, 141–150. IOS Press. https://doi.org/10.3233/atde210092.

Berardinelli, Luca, Stefan Biffl, Emanuel Maetzler, Tanja Mayerhofer, and Manuel Wimmer. 2015. "Model-based co-evolution of production systems and their libraries with AutomationML." In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–8. IEEE. https://doi.org/10.1109/ETFA.2015.7301483.

Breckle, Theresa, Jens Kiefer, Stephan Rudolph, Martin Manns, et al. 2017. "Engineering of assembly systems using graph-based design languages." In *DS 87-1 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies, Vancouver, Canada, 21-25.08. 2017*, 519–528. https://www.designsociety.org/download-publication/39551/Engineering+of+assembly+systems+using+graph-based+design+languages.

Drath, Rainer. 2021a. *AutomationML: A Practical Guide.* Walter de Gruyter GmbH & Co KG. https://doi.org/10.1515/9783110746235.

Drath, Rainer. 2021b. *AutomationML: the industrial cookbook.* Walter de Gruyter GmbH & Co KG. https://doi.org/10.1515/9783110745979.

Drath, Rainer, Arndt Luder, Jorn Peschke, and Lorenz Hundt. 2008. "AutomationML-the glue for seamless automation engineering." In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 616–623. IEEE. https://doi.org/10.1109/ETFA.2008.4638461.

Drath, Rainer, Christian Mosch, Stefan Hoppe, Andreas Faath, Erich Barnstedt, Bernd Fiebiger, and Wolfgang Schlögl. 2023. *Diskussionspapier – Interoperabilität mit der Verwaltungsschale, OPC UA und AutomationML.* Technical Report. AutomationML e.V and Industrial Digital Twin Association (IDTA) and OPC Foundation and VDMA. https://www.automationml.org/wp-content/uploads/2023/04/Diskussionspapier-Zielbild-und-Handlungsempfehlungen-fuer-industrielle-Interoperabilitaet-5.3.pdf.

Drath, Rainer, Peter Weber, and Nicolas Mauser. 2008. "An evolutionary approach for the industrial introduction of virtual commissioning." In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, sep. IEEE. https://doi.org/10.1109/etfa.2008.4638359.

Estévez, E, Marga Marcos, Arndt Lüder, and Lorenz Hundt. 2010. "PLCopen for achieving interoperability between development phases." In *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 1–8. IEEE. https://doi.org/10.1109/ETFA.2010.5641359.

FMI. 2023. "FMI standard." https://fmi-standard.org. Accessed: 2023-06-29.

Fraile, Francisco, Raquel Sanchis, Raul Poler, and Angel Ortiz. 2019. "Reference models for digital manufacturing platforms." *Applied Sciences* 9 (20): 4433. https://doi.org/10.3390/app9204433.

Graeser, Olaf, Barath Kumar, Oliver Niggemann, Natalia Moriz, and Alexander Maier. 2011. "AutomationML as a Basis for Offline - And Realtime-simulation - Planning, Simulation and Diagnosis of Automation Systems." In *International Conference on Informatics in Control, Automation and Robotics*, IEEE. https://doi.org/10.5220/0003537403590368.

Gunnarsson, Sara. 2016. "Evaluation of FMI-based workflow for simulation and testing of industrial automation applications." *Master Thesis* https://www.lu.se/lup/publication/8776878.

Hankel, Martin, and Bosch Rexroth. 2015. "The reference architectural model industrie 4.0 (rami 4.0)." *Zvei* 2 (2): 4–9. https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2015/april/Das_Referenzarchitekturmodell_Industrie_4.0__RAMI_4.0_/ZVEI-Industrie-40-RAMI-40-English.pdf.

Henßen, Robert, and Miriam Schleipen. 2014. "Interoperability between OPC UA and AutomationML." In *8th International Conference on Digital Enterprise Technology - DET 2014 Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution*, Vol. 25, 297–304. https://doi.org/10.1016/j.procir.2014.10.042.

Hoernicke, Mario, Alexander Fay, and Mike Barth. 2015. "Virtual plants for brown-field projects." In *2015 IEEE 20th Conference on Emerging Technologies &amp Factory Automation (ETFA)*, sep. IEEE. https://doi.org/10.1109/etfa.2015.7301462.

Hoffmann, Peter, Reimar Schumann, Talal MA Maksoud, and Giuliano C Premier. 2010. "Virtual commissioning of manufacturing systems a review and new approaches for simplification." In *ECMS*, 175–181. Kuala Lumpur, Malaysia. https://doi.org/10.7148/2010-0175-0181.

Holm, Thomas, Lars Christiansen, Markus Göring, Tobias Jäger, and Alexander Fay. 2012. "ISO 15926 vs. IEC 62424—Comparison of plant structure modeling concepts." In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 1–8. IEEE. https://doi.org/10.1109/ETFA.2012.6489662.

Jackson, Chad. 2019. "What is FMI? How is it related to Virtual Commissioning?" https://virtualcommissioning.com/what-is-fmi-how-is-it-related-to-virtual-commissioning. Accessed: 2023-06-29.

Jackson, Chad. 2020. "Digital Twins of Production Systems: 1D? 3D? What is the Best Fit?" https://virtualcommissioning.com/digital-twins-of-production-systems-1d-3d-what-is-the-best-fit. Accessed: 2023-06-29.

Kaiser, Benjamin, Alexander Reichle, and Alexander Verl. 2022. "Model-based automatic generation of digital twin models for the simulation of reconfigurable manufacturing systems for timber construction." In *Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022*, Vol. 107, 387–392.

https://doi.org/10.1016/j.procir.2022.04.063.

Kiesel, Markus, Philipp Klimant, Nicolai Beisheim, Stephan Rudolph, and Matthias Putz. 2017. "Using Graph-based Design Languages to Enhance the Creation of Virtual Commissioning Models." In *Complex Systems Engineering and Development Proceedings of the 27th CIRP Design Conference*, Vol. 60, 279–283. https://doi.org/10.1016/j.procir.2017.01.047.

Lattanzi, Luca, Roberto Raffaeli, Margherita Peruzzini, and Marcello Pellicciari. 2021. "Digital twin for smart manufacturing: a review of concepts towards a practical industrial implementation." *International Journal of Computer Integrated Manufacturing* 34 (6): 567–597. https://doi.org/10.1080/0951192X.2021.1911003.

Li, Huaxia, Long Tian, and Birgit Vogel-Heuser. 2019. "Automatic Synchronization of Mechanical CAD Models and a SysML-based Mechatronic Model using AutomationML." In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, oct. IEEE. https://doi.org/10.1109/smc.2019.8913908.

Li, Yujiang, Mikael Hedlind, Torsten Kjellberg, and Gunilla Sivard. 2015. "System integration for kinematic data exchange." *International Journal of Computer Integrated Manufacturing* 28 (1): 87–97. https://doi.org/10.1080/0951192X.2014.941937.

Lüder, Arndt, and Nicole Schmidt. 2017. "AutomationML in a Nutshell." In *Handbuch Industrie 4.0 Bd. 2*, 213–258. Springer. https://doi.org/10.1007/978-3-662-53248-5_61.

Lüder, Arndt, Nicole Schmidt, and Ronald Rosendahl. 2015. "Data exchange toward PLC programming and virtual commissioning: Is AutomationML an appropriate data exchange format?" In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, jul. IEEE. https://doi.org/10.1109/indin.2015.7281783.

Lüder, Arndt, Nicole Schmidt, Ronald Rosendahl, and Michael John. 2014. "Integrating different information types within AutomationML." In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 1–5. IEEE. https://doi.org/10.1109/ETFA.2014.7005275.

Martinez, Gerardo Santillan, Seppo A. Sierla, Tommi A. Karhela, Jari Lappalainen, and Valeriy Vyatkin. 2018. "Automatic Generation of a High-Fidelity Dynamic Thermal-Hydraulic Process Simulation Model From a 3D Plant Model." *IEEE Access* 6: 45217–45232. https://doi.org/10.1109/access.2018.2865206.

Metzner, Maximilian, Felix Albrecht, Michael Fiegert, Bastian Bauer, Susanne Martin, Engin Karlidag, Andreas Blank, and Jörg Franke. 2022. "Virtual training and commissioning of industrial bin picking systems using synthetic sensor data and simulation." *International Journal of Computer Integrated Manufacturing* 35 (4-5): 483–492. https://doi.org/10.1080/0951192X.2021.2004618.

Müller, Timo, Simon Kamm, Andreas Löcklin, Dustin White, Marius Mellinger, Nasser Jazdi, and Michael Weyrich. 2022. "Architecture and knowledge modelling for self-organized reconfiguration management of cyber-physical production systems." *International Journal of Computer Integrated Manufacturing* 0 (0): 1–22. https://doi.org/10.1080/0951192X.2022.2121425.

Nagorny, Kevin, Sebastian Scholze, Armando Walter Colombo, and José Barata Oliveira. 2020. "A DIN Spec 91345 RAMI 4.0 compliant data pipelining model: An approach to support data understanding and data acquisition in smart manufacturing environments." *IEEE Access* 8: 223114–223129. https://doi.org/10.1109/ACCESS.2020.3045111.

Oppelt, Mathias, Gerrit Wolf, Oliver Drumm, Benjamin Lutz, Markus Stöß, and Leon Urbas. 2014. "Automatic Model Generation for Virtual Commissioning based on Plant Engineering Data." *IFAC Proceedings Volumes* 47 (3): 11635–11640. https://doi.org/10.3182/20140824-6-za-1003.01512.

Park, Hyeong-Tae, Jong-Geun Kwak, Gi-Nam Wang, and Sang C. Park. 2009. "Plant model generation for PLC simulation." *International Journal of Production Research* 48 (5): 1517–1529. https://doi.org/10.1080/00207540802577961.

Prat, Sophie, Jérémy Cavron, Djamal Kesraoui, Philippe Rauffet, Pascal Berruet, and Alain Bignon. 2017. "An Automated Generation Approach of Simulation Models for Checking

Control/Monitoring System." *IFAC-PapersOnLine* 50 (1): 6202–6207. https://doi.org/10.1016/j.ifacol.2017.08.1014.

Pratt, Michael J. 2001. "Introduction to ISO 10303—the STEP Standard for Product Data Exchange." *Journal of Computing and Information Science in Engineering* 1 (1): 102–103. https://doi.org/10.1115/1.1354995.

Schamp, Matthias, Steven Hoedt, Arno Claeys, El-Houssaine Aghezzaf, and Johannes Cottyn. 2018. "Impact of a virtual twin on commissioning time and quality." *IFAC-PapersOnLine* 51 (11): 1047–1052. https://doi.org/10.1016/j.ifacol.2018.08.469.

Schopper, Dominik, Karl Kübler, Stephan Rudolph, and Oliver Riedel. 2021. "EIPPM—The Executable Integrative Product-Production Model." *Computers* 10 (6): 72. https://doi.org/10.3390/computers10060072.

Schweichhart, Karsten. 2016. "Reference architectural model industrie 4.0 (rami 4.0)." https://img5.custompublish.com/getfile.php/3901260.2265.akzillql7uuipz/RAMEI+4.0.pdf.

Schyja, Adrian, Matthias Bartelt, and Bernd Kuhlenkötter. 2014. "From Conception Phase up to Virtual Verification Using AutomationML." *Procedia CIRP* 23: 171–177. https://doi.org/10.1016/j.procir.2014.10.067.

Süß, Sebastian, Anton Strahilov, and Christian Diedrich. 2015. "Behaviour simulation for virtual commissioning using co-simulation." In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–8. IEEE. https://doi.org/10.1109/ETFA.2015.7301427.

Tao, Fei, He Zhang, Ang Liu, and Andrew YC Nee. 2018. "Digital twin in industry: State-of-the-art." *IEEE Transactions on industrial informatics* 15 (4): 2405–2415. https://doi.org/10.1109/TII.2018.2873186.

Thongnuch, M Sc Suthida. 2021. "An approach to generating high-fidelity models for the virtual commissioning of specialized production machines and cells using MCAD models." PhD diss., Universitätsbibliothek der HSU/UniBwH. http://dx.doi.org/10.24405/13877.

Thongnuch, Suthida, and Alexander Fay. 2017. "A practical simulation model generation for virtual commissioning." In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 1077–1082. IEEE. https://doi.org/10.1109/AIM.2017.8014162.

Thongnuch, Suthida, Alexander Fay, and Rainer Drath. 2018. "Semi-automatic generation of a virtual representation of a production cell." *at-Automatisierungstechnik* 66 (5): 372–384. https://doi.org/10.1515/auto-2017-0108.

Ugarte, Miriam, Leire Etxeberria, Gorka Unamuno, Jose Luis Bellanco, and Eneko Ugalde. 2022. "Implementation of Digital Twin-based Virtual Commissioning in Machine Tool Manufacturing." In *3rd International Conference on Industry 4.0 and Smart Manufacturing*, Vol. 200, 527–536. https://doi.org/10.1016/j.procs.2022.01.250.

Wang, Xi Vincent, and Xun W Xu. 2015. "A collaborative product data exchange environment based on STEP." *International Journal of Computer Integrated Manufacturing* 28 (1): 75–86. https://doi.org/10.1080/0951192X.2013.785028.

Wardhani, Rivai, and Xun Xu. 2016. "Model-based manufacturing based on STEP AP242." In *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, aug. IEEE. https://doi.org/10.1109/mesa.2016.7587187.

Westkämper, Engelbert, Thomas Baudisch, Wolfgang Schlögl, and Gernot Frank. 2012. "Automatic model generation for virtual commissioning of specialized production machines." *Softwaretechnik-Trends: Vol. 32, No. 2* https://doi.org/10.1007/bf03323491.

Xu, Lida, Eric Xu, and Ling Li. 2018. "Industry 4.0: state of the art and future trends." *International journal of production research* 56 (8): 2941–2962. https://doi.org/10.1080/00207543.2018.144480.

Xu, Xun, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. 2021. "Industry 4.0 and Industry 5.0—Inception, conception and perception." *Journal of Manufacturing Systems* 61: 530–535. https://doi.org/10.1016/j.jmsy.2021.10.006.

Yemenicioglu, Ender. 2016. "Data exchange for the physics-based simulation of material handling systems in the digital factory." PhD diss., Dissertation, Magdeburg, Universität, 2016.

http://dx.doi.org/10.25673/4588.

Yli-Ojanperä, Matti, Seppo Sierla, Nikolaos Papakonstantinou, and Valeriy Vyatkin. 2019. "Adapting an agile manufacturing concept to the reference architecture model industry 4.0: A survey and case study." *Journal of industrial information integration* 15: 147–160. https://doi.org/10.1016/j.jii.2018.12.002.

Zhang, Haijun, Qiong Yan, and Zhenghua Wen. 2020. "Information modeling for cyber-physical production system based on digital twin and AutomationML." *The International Journal of Advanced Manufacturing Technology* 107 (3-4): 1927–1945. https://doi.org/10.1007/s00170-020-05056-9.

Zhao, Jiaqi, El-Houssaine Aghezzaf, and Johannes Cottyn. 2023. "A framework for evaluating a generic virtual commissioning data model." In *CIRP Design Conference*, Sydney, Australia. Elsevier. (In Publishing), https://drive.google.com/file/d/1bvyH_E5YNV3BaT037pgCAO897EeGBt09.

Zhao, Jiaqi, Matthias Schamp, Steven Hoedt, El-Houssaine Aghezzaf, and Johannes Cottyn. 2021. "AutomationML in Industry 4.0 Environment: A Systematic Literature Review." *Advances in Automotive Production Technology–Theory and Application* 162–169. https://doi.org/10.1007/978-3-662-62962-8_19.

ZVEI. 2022. "Discussion Paper: Electrical Drive Design with the Digital Twin, Standardised Submodel of the Asset Administration Shell." https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2022/Mai/Diskussionspapier_Digital_Engineering/Diskussionspapier_Digital_Engineering_final.pdf.