Open semantic meta-model as a cornerstone for the design, engineering and management of CPS-based Factories

Stephan Weyer*, Torben Meyer, Jumyung Um*, Moriz Ohmer***

** German Research Centre for Artificial Intelligence (DFKI GmbH) | Kaiserslautern, Germany*
*** Volkswagen AG, Smart Production Lab | Wolfsburg, Germany*

Abstract: Simulation tools are a well-defined and accepted methodology for designing, planning, implementation and operation of systems within the production and will assume a much more decisive role in future factory environments. The growing importance of maintaining digital information all along the production life-cycle across a varied set of tools implies the need a common standard for CPS information models and interfaces. According to the principle of the digital twin concept this will be more important than ever for any exchange of data between different types of *Cyber-Physical Systems* (CPS) and simulation tools. This paper presents an open semantic meta-model which describes relevant functional characteristics of *CPS*-based objects and allows data to be enriched and used as needed for each phase from the object’s design to its integration and operation in an industrial production environment. The CPS template can be used to describe each CPS-based device, workstation, production module or plant throughout its entire lifecycle and will be used along the simulation process for a simpler and more consistent data flow and a less time-consuming process of modelling and model maintenance.

Keywords: cyber-physical systems, digital manufacturing, digital twin, modeling, simulation, smart factory, industry 4.0, internet of things

INTRODUCTION

Computer and communication capabilities will soon be embedded in all types of objects and structures in the physical environment (Rajkumar et al., 2010) and transform those into so-called *Cyber-Physical Systems* (CPS). They are also revolutionizing the manufacturing engineering sector. *Industrie 4.0* is synonym for this transformation of today's factories into smart factories, which are intended to overcome the current challenges of shorter product lifecycles, highly customized products and stiff global competition (Weyer et al., 2015). Machines and devices are becoming intelligent, which means that field devices, machines, production modules and products will be autonomously exchanging information, triggering actions and controlling each other independently (Lee, Bagheri, Kao, 2016).

CPS are key in overcoming the currently rigid planning and production processes and to achieve significantly higher flexibility, adaptability and transparency of production systems (Broy, Kargermann, Achatz, 2010). The traditional production hierarchy will be replaced by a decentralized self-organization enabled by CPS (Zamfirescu et al. 2014). By transferring plug-and-play principles to the industry, CPS enable dynamic adjustments, rearrangement or reengineering processes as needed and mass customization is becoming possible (Gorecky et al., 2016), (Junker, Vorderer, 2016).

However, in terms of engineering a fundamental issue will be maintaining the relevant digital information all along the production life-cycle across a varied set of tools, allowing data to be enriched and used as needed for each phase. As shown in Figure 1, simulations can cover a wide range of applications along the production life-cycle – from the early stages (e.g. layout planning, electrical planning, robot simulation) to the ramp-up and production (e.g. virtual commissioning).

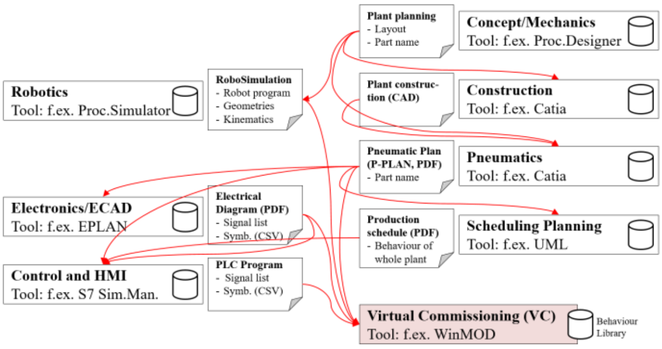

Fig. 1: Example of data flow for Virtual Commissioning

Virtual commissioning especially requires data out of various engineering processes (Weyer et al., 2016). Engineering and simulation is currently characterized by inefficient and time consuming procedures for the development and maintenance of simulation models.

In order to make the design, engineering and management of future CPS-based factories a success, a common standard for CPS information models and interfaces, according to the digital twin concept, is needed for any exchange of data between different types of CPS-based devices and engineering tools (European Commission, 2014). This paper proposes an open semantic meta-model which covers relevant functional characteristics of CPS-based objects. The meta-model will be used to describe each CPS-based device, workstation, production module or plant throughout its entire lifecycle to enable less time-consuming modelling and model maintenance processes along the production life-cycle (Figure 2).

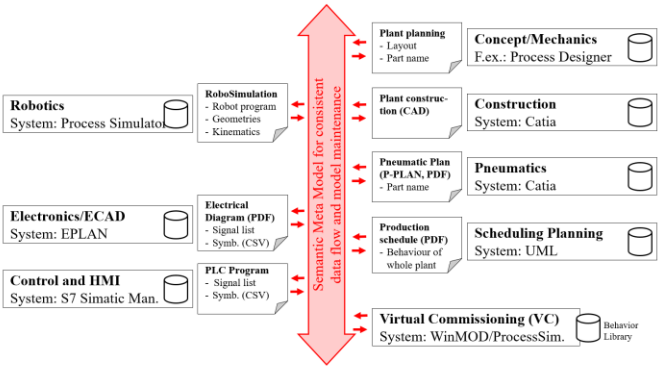

Fig. 2: Semantic meta model for consistent data flow and model maintenance along the production life-cycle

REQUIREMENTS OF SEMANTIC META-MODEL

The following chapter covers key requirements for the meta model, how it should provide a common data basis from which output models can be derived for further usage and how it will fit into a framework to grant vendor-independent access to any stakeholder along the lifecycle.

Structure and characteristics

The open semantic meta-model is required for representing functional characteristics of a CPS which are relevant from its design to its integration and operation in an industrial production environment. The meta model should also achieve a common understanding of static and dynamic CPS data, properties and interfaces for high interoperability and continuity all along the factory life-cycle, between different types of components and heterogeneous simulation tools used. This includes for example:

- description of real time data
- description of filter, transformation and distribution logics
- electrical planning data
- hierarchical information
- 3D shapes as a set of visual meshes
- description of the simplified collision geometries
- description of the kinematics structure
- patterns for the aggregation and assembly of CPS into high-level plant models
- pneumatic wiring diagrams
- energy data about the general energy consumption
- signals and control data
- relations between CPS

Access, storage and security aspects

The information stored within meta model need to be available for the simulation environment. A binding to the simulation framework is fundamental. It should be possible to navigate from the framework to the data and grant vendor-independent access to planners, manufacturers and other stakeholders to simulation models deposited (easy to query and easy to update).

New simulation results are stored in the semantic repository of the framework and also used by other software tools along the simulation-chains. Furthermore, the semantic meta model should provide the capability to track changes. This encompasses capturing the source of a change well as its content. Additionally, the model should support data security access control.

Output model aspects

The meta model should provide a common data basis, from which output models can be derived for further usage. This includes, for instance, the generation of behavior models e.g. for virtual commissioning. Providing relevant behavior models of new components will simplify and speed up the process of model building e.g. for virtual commissioning and the evaluation process of production planning itself. The model should facilitate the (semi-) automatic generation of simulation models e.g. mechatronic models.

META MODEL FOR CPS

The CPS data structure can be thought as a container that maintains semantic links among different standard descriptions. It has room for specifying new properties and interoperable behavioral models and holds references to physical devices and to communication details. Figure 4 shows the template of a CPS where it is possible to identify five main sections (Figure 3). Each section has a unique identifier.

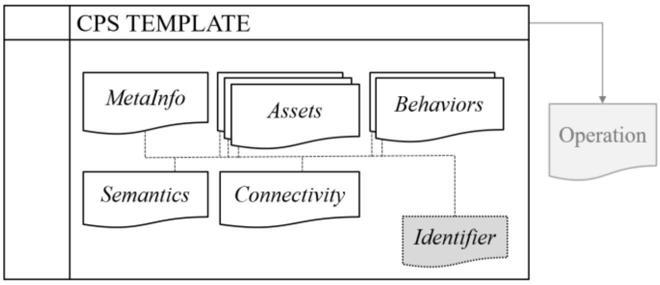

Fig. 3: Main sections of the semantic meta-model

CPS Meta Data and Connectivity

The CPS MetaInfo section contains meta data related to vendor, version, type of CPS, owner etc. The basic data structure of the *ODVA Machinery Information Base Data Structure* serves as a basis to define relevant attributes required to describe fundamental identifying information of automation objects (Beudert, Leurs, Zuponcic, 2015). This section is mandatory for any CPS template. Additionally, the section 'connectivity' contains communication parameters for connecting to reading from or writing to a CPS object. E.g. the section stores the device's IP address and port of the device or it contains the PLC memory location and types of data stored.

CPS assets and behaviors

This section contains references to external resources like models or binary data that these models or simulation tools can use. An important feature that should be supported is linking between runtime properties and properties defined inside assets and between properties defined by two different assets. Assets will fall under a CPS' static data of the CPS because they represent self-contained models that rarely change.

The CPS behavior section contains references to runnable behavioral models that represent functionalities and the operative logics of the physical system as well as raw data stream aggregation and processing functions. Simulation tools are able to use the former to improve the reliability of simulations whereas the latter should run inside a support infrastructure to update the runtime properties of the CPS model. Behavioral models reside in external resources, similarly to assets.

Semantic aspects

This section describes semantic aspects for the classification and clear description of the CPS-based objects. AutomationML defines a set of basic role classes (AutomationMLBaseRoleClassLib) but does not define semantics of production system components themselves. Instead it integrates existing semantic definitions as given for example in the eCl@ss classification standard (AutomationML, 2014).

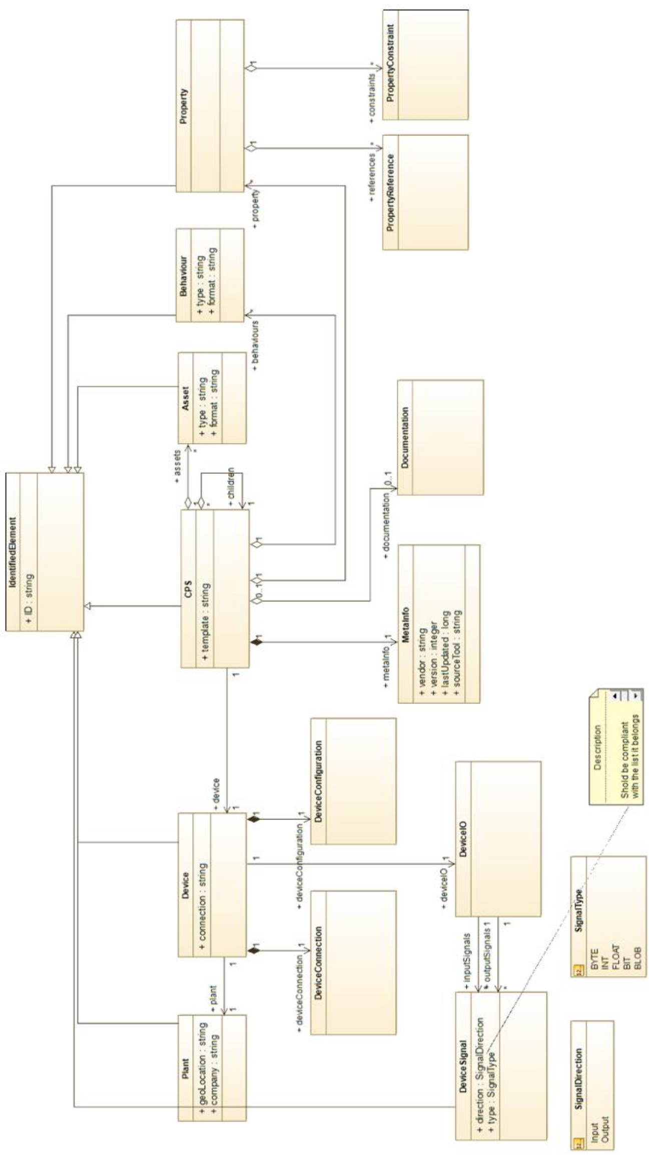

Fig. 4: Meta model for CPS-based objects

The XML schema based data format AutomationML is used to describe parts of the CPS template (CAEX). The table below shows the mapping of meta model classes to AutomationML elements.

Meta Model	AutomationML terminology
CPS Template	SystemUnitClassLib
CPS Instance	InstanceHierarchy, InternalElement
Asset	ExternalDataConnector, COLLADAInterface
Behavior	ExternalDataConnector, PLCopenXMLInterface
Property	Attributes
Device	Role classes: Resource, DiscManufacturingEquipment
Plant	Role classes, ResourceStructure
Documentation	ExternalDataConnector

Table 1: Mapping of meta model classes to AML

In general, a CPS template contains or will be enriched over its life cycle all the models and all properties that simulation tools can use. Ideally, device manufacturers should directly provide CPS templates usable to create digital twins along with physical devices. However, a CPS template is a standalone model that is complete from a digital point of view, but it is still not applied in any plant model. By adopting plug-and-play principles in industrial technologies, the meta model makes a substantial contribution to the digital and virtual integration of the CPS itself. Plug-and-play principles will transform future production lines into highly modular and flexible setups, which can be dynamically adjusted and rearranged any time and without interfering production (Hodek, 2013). The table below shows the individual steps during device integration. Unlike legacy systems, the meta model provides the information needed for each step.

States	Description	Required data	Legacy System	Meta Model
Wait/ Sleep	Just after power up, but not connect with the machine	identification	not used	Identified Element
Isolate	Detects new machine drop it out in list of waiting machines	interface type	not used	Device
Configure	Create new adaptor for new machine	gata scheme	vendor's adaptor	Device Signal/ IO
Visualize/ Engineer	e.g. Load CAD model in 3D viewer	e.g. geometry	manual input	Asset
Simulate/ Control	Edit logic code of the machine controller	e.g. control logic	HMI on machine itself	Behavior

Table 2: Mapping of meta model classes to AML

EVALUATION ON AML SAMPLE MODEL

For preliminary evaluation of the meta model, a conveyor was modelled completely at the relevant level of detail – with focus on virtual commissioning. It is modeled in AutomationML and associated data formats. The model includes a PLC with control logic, the geometry and the signal connections between conveyor, light barriers and the PLC.

Meta Model Classes	Entities of Conveyor	AutomationML types
CPS	Conveyor	SystemUnitClassLib
MetaInfo		
Documentation		ExternalDataConnector
Device	PLC, Engine, LightBarrierStart, LightBarrierEnd	Role classes: Resource
DeviceConnection	PLC-Interfaces Conveyor-interface, Engine	
DeviceConfiguration		
Plant		Role classes: ResourceStructure
Asset	COLLADA, JT	ExternalDataConnector, COLLADAInterface
Behavior	PLCLogicInterface, EngineLogicInterface, MaterialFlow	ExternalDataConnection
Property	Output	
DeviceSignal	Input & Output	
DeviceIO	Input & Output	

Table 3: Mapping of conveyor sample table to the meta model and AML types

Table 3 shows the mapping to the model classes. The conveyor itself was modeled as an object in the AutomationML instance hierarchy and besides some attributes it has a COLLADA interface as well as a JT interface because both options were checked. Moreover, it has two material flow interfaces: one at the beginning at one at the end. Two light barriers were modeled in the same COLLADA file and linked to their own instance element objects as well as their logic interfaces. In this case, the PLC does not have a geometry representation but the Boolean interfaces were modeled as PLCOpen-XML interfaces.

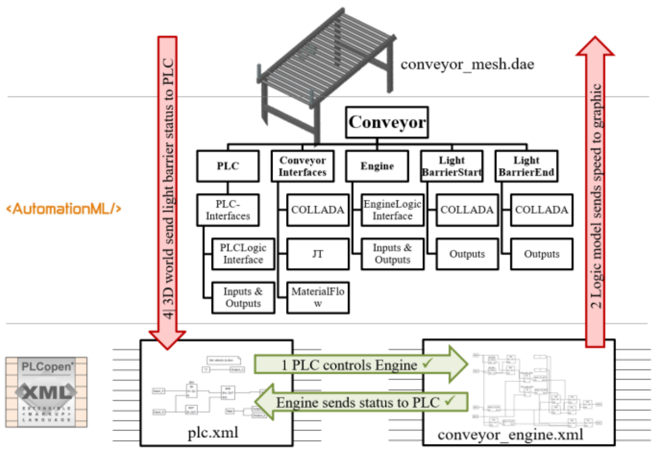

Fig. 5: Sample model for VC in AML

The wiring between PLC, light barrier and conveyor was modeled utilizing internal links between the interfaces of the objects. The PLC object contains logic interfaces that reference an external PLCOpen-XML file (control logic).

The engine object of the conveyor references another external PLCOpen-XML file (behavior logic). Within the PLCOpen-XML files globalID attributes were used as anchor points. The link between the PLC and conveyor engine is modeled, the mechatronic connection between the geometry and the outcome of the logic is still missing. Building on existing work (Meyer, 2014), it will be one focus of future work within the project.

ANNOTATION

This project (MAYA) has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 678556.

REFERENCES

AutomationML (2014). The Glue of Seamless Automation Engineering: Whitepaper AutomationML Part 2 – Role class libraries. AutomationML consortium. Magdeburg, Germany.

Beudert, R.; Leurs, L.; Zuponcic, S. (2015). Machinery Information Base Data Structure; ODVA Industry Conference & 17th Annual Meeting. Frisco, Texas.

Broy, M., Kargermann, H., Achatz, R. (2010): Agenda Cyber Physical Systems: Recommendations for the 8th Framework Program. German Academy of Science and Engineering (acatech). Munich, Germany.

European Commission (Ed., 2014). Pointing out research priorities leading to the next generation of simulation and forecasting technologies: Pathfinder Project. Brussels.

Gorecky, D., Weyer, S., Hennecke, A., Zühlke, D. (2016): Design and Instantiation of a Modular System Architecture for Smart Factories. In: Proceedings of 12th IFAC Workshop on Intelligent Manufacturing Systems (IMS 2016). Austin, USA, IFAC, 2016.

Hodek, S (2013). Methode zur vollautomatischen Integration von Feldgeräten in industrielle Steuerungssysteme: Ein Beitrag zur Plug&play-Feldgeräteintegration. Dissertation, TU Kaiserslautern, Fortschritt-Berichte pak, Bd. 26

Junker, S. and Vorderer, M. (2016). Highly versatile plug&produce assembly systems. 16. Internationales Stuttgarter Symposium. Springer Fachmedien, Wiesbaden.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 3, 18-23.

Meyer, T. (2014). Wirtschaftliche Erstellung von Emulationsmodellen für die Virtuelle Inbetriebnahme. Dissertation. Logos Verlag, Berlin.

Rajkumar, R., Lee, I., Sha, L., Stankovic, J.:Cyber-physical systems: the next computing revlotion. In: Proceedings of the 4th Design Automation Conference, pages 731-736,2010.

Schmidt, N.; Lüder, A. (2015). AutomationML - The Glue for Seamless Automation Engineering: AutomationML in a Nutshell. AutomationML consortium. Magdeburg, Germany.

Weyer, S., Meyer, T., Ohmer, M., Gorecky, D., Zühlke, D. (2016). Future Modeling and Simulation of CPS-based Factories: an Example from the Automotive Industry. In: Proceedings of 12th IFAC Workshop on Intelligent Manufacturing Systems (IMS 2016). Austin, USA, IFAC, 2016.

Weyer, S., Schmitt, M., Ohmer, M., & Gorecky, D. (2015). Towards Industry 4.0-Standardization as the Crucial challenge for highly modular, multi-vendor production systems. IFAC-PapersOnLine, 48(3), 579-584.

Zamfirescu, C.-B.; Pîrvu, B.-C.; Loskyll, M.; Zühlke, D. (2014). Do Not Cancel My Race with Cyber-Physical Systems. IFAC: Promoting automatic control for the benefit of humankind. Cape Town, South Africa.