Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 67 (2018) 128 – 133

www.elsevier.com/locate/procedia

11th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '17

UML2OPC-UA – Transforming UML class diagrams to OPC UA information models

Florian Pauker[a,]*, Sabine Wolny[b], Solmaz Mansour Fallah[a], Manuel Wimmer[b]

[a]*Institute for Production Engineering and Laser Technology, Getreidemarkt 9, 1060 Vienna, Austria*
[b]*Institute of Software Technology and Interactive Systems, TU Wien, Favoritenstraße 9-11/188-3, 1040 Vienna, Austria*

* Corresponding author. Tel.: +43158801311382 ; fax: +4315220131199. *E-mail address:* pauker@ift.at

Abstract

The emergence of cyber physical systems in the manufacturing domain creates new requirements for shop floor devices. Due to their diverse structure, buildup a seamless communication is one major challenge.
In recent years, several communication standards, e.g., OPC-UA, tried to tackle this issue. Until now, they have not fully developed their potential, because of inherent implementation complexity and specific formats, although well-known modeling standards such as UML exist. The presented approach aims to overcome this complexity by enabling an automatic transformation from UML class diagrams to OPC-UA information models using ATL and by extending UML to guarantee information preserving transformations.

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Keywords: UML; OPC UA; ATL; MDE

1. Introduction

Todays' challenge in manufacturing is the transformation of common manufacturing systems into cyber-physical production systems (CPPS). Latter presupposes the integration of so-called cyber physical systems (CPS) in manufacturing equipment [1]. To meet the defined requirements of CPS, they must come with self-X properties. One of those mentioned properties has to be self-organization in order to be capable of communication through networks. CPS need to be aware of themselves (self-awareness) and their communication partners [2]. Latter concludes the urge for self-description and self-protection in order to divide and protect shareable from not shareable information [2], [3].

Realizing CPS means, that sensors, machines and manufacturing systems are connected with other real and virtual objects or processes, e.g., ERP and MES systems. All those systems need a semantic rich description. There are several different available interfaces and communication standards, which are feasible for realizing the requirements [4]. This diversity and the missing standardization is one reason why the interoperability between different components is challenging.

With Ethernet there is already a standard for communication on the physical layer available, which is considered as the main foundation for standardized communication on higher levels. With OPC Unified Architecture (OPC UA), the latest specification of the OPC standard, there is a technology available, which allows connecting heterogeneous systems on different factory levels in a standardized way [5], [6].

OPC UA is standardized since 2011 (IEC 62541), but its adoption in the manufacturing domain is still in its infancy - although the last decade has been filled with new appealing debates about the role of OPC UA in this domain [7]. On the one hand, this trend is encouraged through the need for standardized domain specific models (companion specifications), and on the other hand, through the shortage of tools to support the model creation, implementation and maintenance.

Domain Specific Languages (DSL) seem promising to reduce the complexity of OPC UA application development. Especially for modelling the Address space and

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering
doi:10.1016/j.procir.2017.12.188

implementation of OPC UA metamodel rules [8]. Therefore, we present a Model Driven Engineering (MDE) [9] approach for developing OPC UA information models. This approach extends UML models with profiles for describing CPS-related concerns. One key concept of this approach is the automatic transformation of UML models into OPC UA information models. This paper captures the necessary extension of the UML models, the OPC UA profile, and some core transformation rules for the transformation of the structural concerns of CPS.

The following sections describes the mandatory background, OPC UA and MDE, and some related work in this area. In Section 3, we give an overview of our approach. Mappings and transformation rules from UML to OPC UA and their implementation will be described in Section 4. Finally, Section 5 concludes with an outlook on future work.

2. Prerequisites: MDE; OPC UA, and UML

In this section, we present the main building blocks for our approach at a glance.

2.1. Model Driven Engineering (MDE)

In Model Driven Engineering (MDE), the abstraction power of models is applied to tackle the complexity of systems [9], [10]. MDE follows the principle "everything is a model" for driving the adoption and ensuring the coherence of model-driven techniques, in the same way as the principle "everything is an object" was helpful in driving the object-oriented techniques in the direction of simplicity, generality, and integration [11]. Historically, MDE has been mainly applied in software engineering [9], [11], but in recent years, the application of MDE has been increasing in the industrial automation domain as well [12].

A key principle of MDE is to address engineering with formal models, i.e., machine-readable and processable representations. Based on this foundation, modeling provides a set of advantages for driving the engineering process effectively and efficiently. The application of model validation, testing, verification, simulation, transformation, and execution enables the automation of engineering process steps and support the traceability of engineering artifacts to improve quality management.

In 2001, the OMG adopted the Model Driven Architecture (MDA), as a concrete MDE approach for using models in software development [13]. MDA has three primary goals: *(i)* portability, *(ii)* interoperability and *(iii)* reusability, through architectural separation of concerns.

MDA specifies three default models of a system corresponding to the three MDA viewpoints defined above. These models correspond to layers of abstraction, since within each of these three layers a set of models can be constructed, each one corresponding to a more focused viewpoint of the system (user interface, information, engineering, architecture, etc.).

Model transformations are considered as the key technique to automate software engineering tasks in MDE [9], [10], by providing the essential mechanisms for manipulating models. In fact, they allow transforming models into other models or into code, and are essential for synthesizing systems in MDE. In this paper, we focus on model-to-model (M2M) transformations. Generally, a M2M transformation is a program executed by a transformation engine that takes one or more models as input to produce one or more models as output [9]. An important aspect of model transformations is that they are developed on the metamodel level. Thus, they are reusable for all valid models of the input metamodel.

2.2. OPC Unified Architecture - OPC UA

OPC Unified Architecture (OPC UA) is the latest standard provided by the OPC Foundation and building upon OPC Classic (see figure 1). It is standardized by the IEC 62541 standard series. It enables the secure, reliable and vendor-independent transport of raw data and pre-processed information from sensor and field level up to the control system and into production planning systems. Its multi-layered approach fulfills goals like for example, platform independence. Its platform independence is reasoned by the wide spectrum of hardware platforms and operating systems, which can be used to run OPC UA functions. OPC UA also fulfills the function equivalence to OPC Classic, which includes events, methods, on-demand read and write data, address space and discovery.

OPC UA's combination of two crucial abilities, like adding new features without affecting existing ones (extensibility) and information modeling, are beneficial to capture complex systems.

One of the main reason for the sufficiency of OPC UA in industrial area is its capability of information representation and modeling. The reason is that the metamodel (see figure 3) can be extended with standardized and user specific information models [14], [15]. OPC UA comes with an information-modeling framework that shifts quantities data values into qualitative information. Due to its object-oriented modeling approach, OPC UA gives all the advantages of this paradigm. One of them is the capability to capture complex multi-level structures, to model and to extend them.

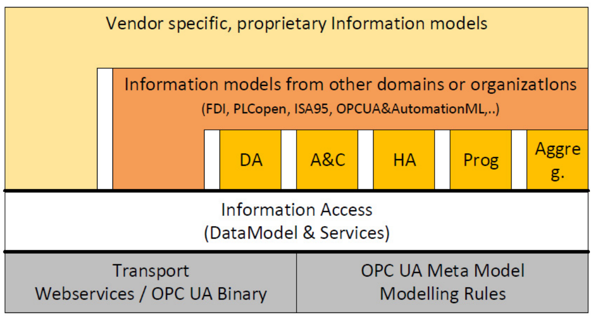

Fig. 1. OPC UA model hierarchy.