Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 60 (2017) 279 – 283

www.elsevier.com/locate/procedia

27th CIRP Design 2017

Using graph-based design languages to enhance the creation of virtual commissioning models

Markus Kiesel[a,*], Philipp Klimant[b], Nicolai Beisheim[a], Stephan Rudolph[c], Matthias Putz[b]

[a]*Albstadt-Sigmaringen University, Jakobstraße 1, 72458 Albstadt-Ebingen, Germany*
[b]*Institute for Machine Tools and Production Processes, Technische Universität Chemnitz, Reichenhainer Straße 70, 09126 Chemnitz, Germany*
[c]*Institute of Statics and Dynamics of Aerospace Structures, Universität Stuttgart, Pfaffenwaldring 27, 70550 Stuttgart, Germany*

* Corresponding author. Tel.: +49-7571-732-9426; fax: +49-7571-732-9152. *E-mail address:* kiesel@hs-albsig.de

Abstract

'Industrie 4.0' based production systems are likely to change the way future products are manufactured. As information technology gains influence on these systems there is a chance of higher flexibility due to decentralized logic and artificial intelligence. All this leads to a higher complexity and also indeterminism is feasible. Therefore simulation technologies will become a mandatory requirement, especially virtual commissioning will get necessary as the amount of software is rising.
A lot of manpower is required to establish and maintain a virtual commissioning system as it needs a large database of standard components. Therefore in most cases small- and mid-sized companies are forced to avoid such technologies. Using graph-based design languages to create virtual commissioning models can help to solve this problem. The basic principle is to shape an abstract model of a production system which will then be individually built within the domain specific tools. One of these should be a virtual commissioning tool to evaluate the functionality of the built model. If a change in the design is necessary, the new virtual commissioning model can be regenerated automatically. This approach is even more reasonable, if graph-based design languages are used throughout the whole product life cycle.

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference

Keywords: Simulation; Design method; Design optimization; Digital Manufacturing System

1. Motivation

Virtual Commissioning is a key technology to increase robustness and quality of production systems, by providing the opportunity to test, evaluate and enhance processes within the system, based on a digital mock-up (see e.g. [1], [2]).

Currently only big engineering teams can use this technology because of the high manpower required to build the simulation models. With technologies like *Industrie 4.0* or *Internet of Things* (short: IOT) on the rise, the flexibility of production systems will increase (see e.g. [3], [4], [5]). But this also causes a huge complexity growth of the production systems, e.g. caused by the decentralized logic and the artificial intelligence.

Hence the usage of technologies like virtual commissioning will get mandatory, even for small- and mid-sized companies, to maintain or expand their market share. As mentioned earlier, currently the biggest issue is to create and update the simulation models needed for the virtual commissioning process. With the increasing complexity this issue even gets bigger. One possible solution is an automated creation of these simulation models, by applying graph-based design languages.

2. Graph-based Design Languages and AutomationML

The methodology stated in this paper is based on two main concepts to enhance the creation of virtual commissioning models. This chapter will provide a general overview of these concepts.

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
doi:10.1016/j.procir.2017.01.047

2.1. Graph-based Design Languages

Graph-based design languages are used for design process automation based on abstract knowledge in order to get the best possible result (see e. g. [6], [7]). A design language consists of three fundamental components:

- Vocabulary
  The vocabulary is one part of the abstract knowledge. It describes the available components within the design language and their correlation.

- Rules
  The rules are the second part of the abstract knowledge and are basically the blueprint for the design process. The rules can be influenced by a variety of parameters, which can lead to completely different models.

- Compiler
  A compiler instantiates the design described by the vocabulary and rules. The model that was generated by the compiler can be further processed as meta-model for several other applications.

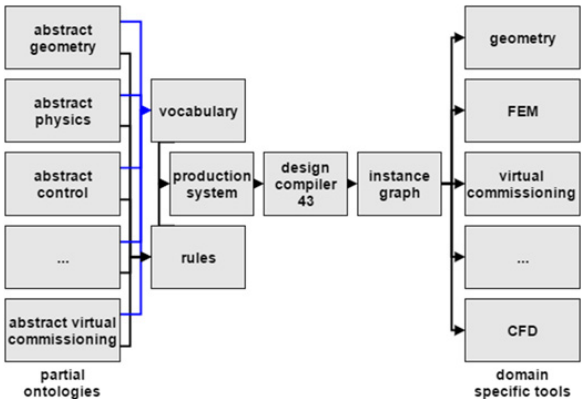

Figure 1: Overview of graph-based design languages [8]

Graph-based design languages can form nearly every aspect of the target model, for instance the geometry of a product or the program code used for machining (CNC-code). The necessary abstract knowledge to form these aspects has to be introduced to the system by experts in their field (left-hand side in the figure above). A complete overview of the correlations within a graph-based design language process is also shown in Figure 1.

In this research project the creation and execution functionality of graph-based design languages is provided by the *Design Compiler 43* (IILS mbH [9]). It has already proven the benefits of graph-based design languages in several research (e.g. [10], [11]) and business projects.

2.2. AutomationML

Due to the rising complexity of *Industrie 4.0* based production systems it is obligatory that engineering teams of different departments can exchange information efficiently. One format which can handle heterogeneous data is the XML-based data format AutomationML (see e. g. [12], [13]). It can contain much more information than for example a typical CAD exchange format like STEP or IGES. To make AutomationML easy accessible it incorporates several standards.

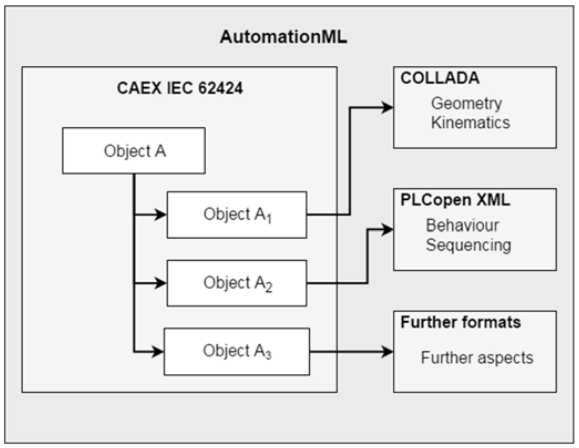

Figure 2: AutomationML Overview [14]

The open standards, which are used by AutomationML, are shown in Figure 2. The AutomationML file itself is based on the CAEX Format (IEC 62424) which is just slightly enriched. As it is XML-based and due to the possibility to reference other files, it is easy expandable. The present components, the hierarchical structure as well as the connection between the components are described with the CAEX Format.

The COLLADA standard provides the functionality for the representation of geometry. It is capable of saving geometry as a boundary representation (typically for CAD software) as well as a triangulated mesh representation. Besides the geometry, COLLADA can also contain information about the kinematics and physics of an object, as well as other geometry related information.

The PLCopen XML format is also included into AutomationML and makes it especially interesting for virtual commissioning purposes. Since it is based on the IEC61131-3, it adds the functionality to store and transfer programming languages for PLCs, embedded controls and industrial PCs. This data can be evaluated on software or hardware in the loop systems typically required for virtual commissioning.

Also shown in Figure 2 is the ability to incorporate further formats to add special functionality to AutomationML.