PCFF: Plug & Control for flexible transport equipment based on AutomationML

Miriam Schleipen
Fraunhofer IOSB
Fraunhoferstr. 1,
76131 Karlsruhe, Germany
miriam.schleipen@iosb.fraunhofer.de

Michael Okon
Fraunhofer IOSB
Fraunhoferstr. 1,
76131 Karlsruhe, Germany
michael.okon@iosb.fraunhofer.de

Thomas Hövelmeyer
cjt Systemsoftware AG
Am See 14a,
76297 Stutensee, Germany
thomas.hoevelmeyer@cjt.de

Andreas Wagner
cjt Systemsoftware AG
Am See 14a,
76297 Stutensee, Germany
andreas.wagner@cjt.de

Gerhard Wolff
cjt Systemsoftware AG
Am See 14a,
76297 Stutensee, Germany
gerhard.wolff@cjt.de

Halit Demir
Gebhardt Fördertechnik GmbH
Neulandstr. 28,
74889 Sinsheim, Germany
h.demir@gebhardt-foerdertechnik.de

Matthias Jentsch
Gebhardt Systems GmbH
Neulandstr. 28 / 1,
74889 Sinsheim, Germany
m.jentsch@gebhardt-group.com

Kai Furmans
Karlsruher Institut für Technologie (KIT)
Gotthard-Franz-Straße 8,
76131 Karlsruhe, Germany
kai.furmans@kit.edu

Dennis Asi
Karlsruher Institut für Technologie (KIT)
Gotthard-Franz-Straße 8,
76131 Karlsruhe, Germany
dennis.asi@kit.edu

Abstract

'Industrie 4.0' summarizes ideas, technologies, and methods to equip production resources, processes, and methods for future. One challenge is to deal with the growing amount of individualization of products [1]. Today modern production processes are monitored and controlled via process control systems or additional MES (Manufacturing Execution Systems. This results in a significant amount of engineering effort for these systems. Engineering is a creative process which involves many engineers, their knowledge, and experience. At the moment this complex activity is done manually and often results in high costs and a number of errors. Goal of the research project 'PCFF' of GEBHARDT Fördertechnik GmbH, cjt Systemsoftware AG, Fraunhofer IOSB, and KIT/IFL is to step towards 'Industry 4.0' in this context.

The project started in 2013, ends in 2015, and is funded by the Federal Ministry for Economic Affairs and Energy (BMWi) (FKZ KF2074712KM2).

The present paper discusses the basic concept and architecture and highlights first project results.

1 Introduction, Motivation and Goal

Parts of the plant engineering phase can be automated for flexible, adaptive and standardized conveyor modules. This varies from the fusion of existing data from the plant planning phase to a consistent generation of visualization for different monitoring and control systems.

For this purpose, an automation framework for a rapid, cost effective engineering, monitoring and control of production plants is needed (especially for transport in intralogistics).

Requirements therefore are modular conveyor units, which can be put together according to a modular system principle. Chapter 2 deals with conveyor modules and their features and functions. Not only the relevant software interfaces are necessary, also the hardware connectors have to support the modularity (see Figure 1, 1).

These individual conveyor modules should be put together to a complex conveying system with minimal effort and can then directly taken into operation. Through this procedure, failures and the duration of shutdown are reduced respectively minimised.

For this solution the conveyor modules are developed as a basis, which are provided with a special control box (see Figure 1, 2). When the conveyor modules are put together to a conveyor system, the decentralised control algorithm recognises the topology automatically.

The evaluation is realized by a demonstration plant which is shown in chapter 3.

The principle of Plug & Control for flexible conveyor technique (PCFF) is described in chapter 4. It is based on two IEC-Standards, AutomationML and OPC UA, which are combined. OPC UA [2] is a platform independent standard series (IEC 62541) for the communication of industrial automation devices and systems. OPC UA is important for the industrial communication (also Machine-to-Machine) and the data exchange in the production environment. OPC UA is based on a service-oriented architecture (SOA) and is the successor of the classic OPC, which is nowadays used very often in the production.

AutomationML [3] is an open XML-based standard series (IEC 62714) for describing and modelling production plants and components. The goal of the standard is to develop a neutral data format, which bridges the today existing gap between engineering tools of the digital factory and the automation engineering.

The topology data including the information about the individual conveyor modules are exported and then converted automatically in a corresponding process control system. For this the data from engineering (information about mechanics, electronics, topography, topology, structure, and connection logic) are merged to an AutomationML file/description (see Figure 1,3). The import step is explained in chapter 5.

Based on this AutomationML file, corresponding images and project data are generated (see Figure 1, 4) and exported in the requested tool format (control system, MES-component, etc.) (see Figure 1,5). The export in the tool format is described in chapter 6.

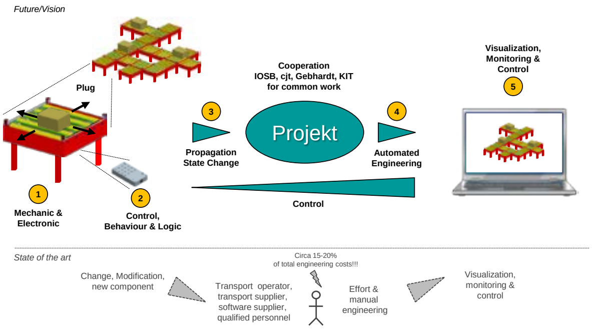

Figure 1. From plant to visualization

The implementation of this concept is carried out by the Gebhardt FlexConveyor, the KIT-control box, the production monitoring and control system Siemens WinCC [4], ProVis.Visu® [5] and the proprietary Gebhardt visualization components. The enhancement for Android-App is intended and considered in the architecture. Production monitoring and control is carried out by OPC UA (Unified Architecture) [6]. First results are described with the help of the example plant chapter 7.

2 Gebhardt conveyor modules

The modular conveyor modules by Gebhardt implement the Plug & Play principle for industry use and thus allows a high flexibility and a high degree of reuse. They support both the container and box conveyor technology as well as the pallet conveyor technology. They consist of a modular system of predefined, standardized conveyor modules. There exist 14 conveyor modules and each of them belongs to a predefined type series. The type series represents different lengths, widths and required angles in a predefined size gradation. The conveyor modules are built in a modular way, i.e. the same attaching parts can be used in every size gradation.

In order to assemble production lines from the 14 conveyor models and their type series (see Figure 2), the conveyor modules also have to be modular to the outside, so that they can be combined together.

Figure 2. Example of plant consisting of Gebhardt conveyor modules

In order to achieve this, standardized mechanical and electrical interfaces have been developed.

For a rapid and simple modification, the mechanical connection technology is realised in such a way that the connection of two conveyor modules can be carried out without tools. They can be telescoped and bolted (see Figure 3, left side).

The communication between the modules is based on Ethernet. Every conveyor module includes a special control box which is able to recognize the topology of the ad hoc combined conveyor modules by a decentralized control algorithm. It forwards the topology to the control system. Subsequently the operating mode and the material flow can be configured by the process control system. In a simple operating mode load carriers are moved along static transport routes. Furthermore distributions can be predefined for e.g. switches. In an advanced operating mode autonomous transport is possible. Identified load carries (e.g. via barcode or RDIF scanner) are transferred into the system and are transported to the identified destination via a generated deadlock-free route (see Figure 3, left side).

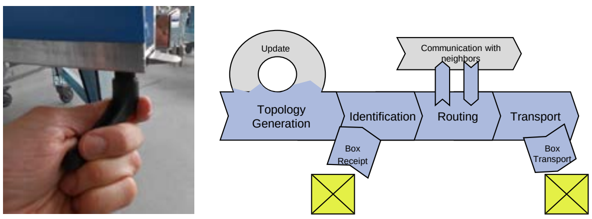

Figure 3. Tool-free plugging method (left), advanced operating mode for autonomous material flow (right)

3 Example plant for evaluation

The entire PCFF-process should be executed and evaluated by means of two topologies including all steps from topology recognition of the plant to the automated engineering. Layout 1 (Figure 4, left side) shows a simple cyclic topology with an accurately defined conveyor direction (clockwise/anticlockwise). Layout 2 (Figure 4, right side) shows a cyclic layout with several switches leading to inclusions and exclusions. The material flow runs in an accurately defined conveyor direction (clockwise/anticlockwise).

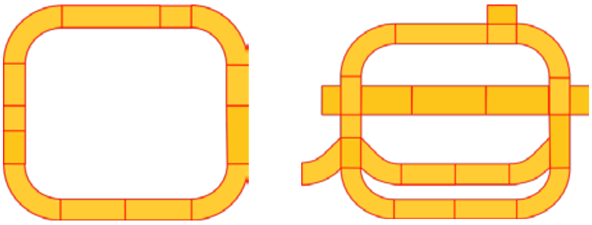

Figure 4. Layout 1 – cyclic topology (left), Layout 2 – extended cyclic topology (right)

4 General architecture

Within the framework of PCFF, two different phases must be differentiated – the offline or project planning phase and the online or monitoring phase. The corresponding software for this purpose was initially implemented as a Framework for manual exchange. An extension for an automated process and exchange should be implemented later on. The two phases are described in the following.

4.1 Offline phase

The offline phase (see Figure 5) consists of the data fusion (topology, topography, structure, geometry, IO-connection) into a standardized, neutral exchange format followed by the generation of process visualization in different views for different visualization systems. This takes place within the 'PCFF-Fusion-Assistant'. AutomationML [7] as plant data exchange format allows modelling all relevant data. Thus it is dedicated to achieve this goal.

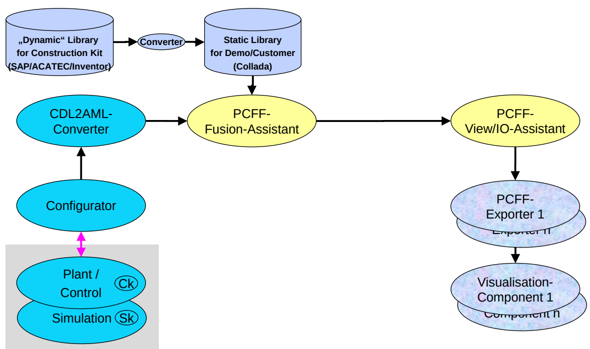

Figure 5. Framework „Offline“

After the storage of all relevant information in AutomationML, the ‚PCFF-View/IO-Assistant' (PCFF-Configurator) exports all information into proprietary XML, COLLADA, and BMP files. These descriptions now include all information relevant and reduced for monitoring and control systems, especially hierarchies depicted by different images, process connections and view-based (e.g. Top, ISO) graphical information in 2D. These descriptions are independent from the final system.

Subsequently the content of these files is exported by a 'PCFF-Adapter' which is specific for the different monitoring and control systems (e.g. Siemens WinCC/Provis.Visu®/Gebhardt). This step also consists of the IO-connection information and includes the usage of the system specific description format or system specific programming interfaces. After the import into the monitoring and control systems, the 'look and feel', as well as the information and usage concept, in all the systems is quite similar.

4.2 Online phase

The online phase comprises (see Figure 6) the possibility of monitoring and control by means of a standardized, neutral communication protocol based on different views in the process visualization for diverse systems. OPC UA [6] allows communicating process information (e.g. motor speed, transport direction, or control commands) and represents a standardized communication and management interface from software systems to the production process. Thus it is dedicated to this goal.

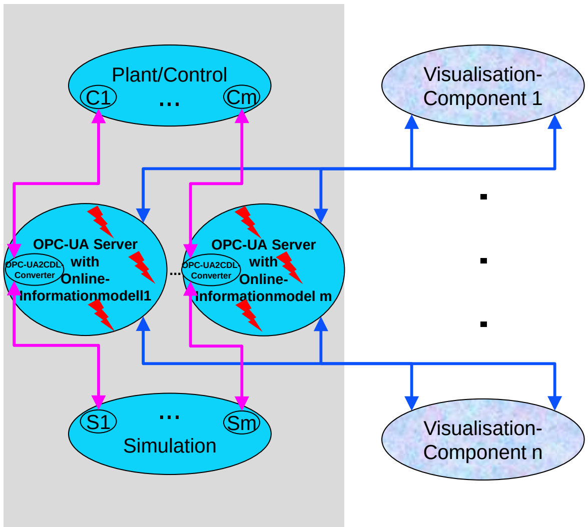

Figure 6. Framework „Online“

5 Import and fusion

To be able to use data from different data sources (e.g. from different planning tools of different planning steps), this data has to be integrated in a common model. As already described, AutomationML was chosen for this purpose. The plant component model consists of different parts which are connected by strictly typed links[1]:

- Topology (attributes and relations of objects in their hierarchical plant structure) implemented by means of CAEX (Computer Aided Engineering Exchange),
- Geometry (graphical attributes and 3D information) implemented by means of COLLADA (COLLAborative Design Activity),
- Kinematic (relations and dependencies of objects to describe movements) implemented by means of COLLADA,
- Logic (process sequences, internal behaviour, and IO-connections) implemented by means of PLCopenXML.

All this information already exists during the planning of transport equipment; however it is distributed to different tools and data management systems. Thus, the information must be collected, harmonized and unified within a common model which forms the base for further processing steps for production monitoring and control.

Figure 7 depicts the data flow and formats to the point of an AutomationML file with all necessary information for the generation of process visualization by the 'PCFF-View/IO-Assistant' (PCFF-Configurator).

As the provided data does not correspond to the AutomationML format, different converting tools were implemented for the conversion tasks. The converted files are then fusioned to one single AutomationML file which references to the semantically lifted graphical information in the COLLADA format.

The following import information forms the base for the fusion step:

- CDL files (Conveyor Description Language): Proprietary format for the description of conveyor modules and plants. These files include information about mechanics, electronics, topology, structure, a connection table to other decentral conveyor modules, and IO-connections to the process.
- Transformation rules (CDL → AML): These rules describe the transformation from the CDL to the AutomationML format including

[1] PCFF only uses the first two types.

types, instance hierarchies, groups, topography, graphical references, and the connection to the process.

- DAE files (COLLADA): Standardized format for the description of graphical information. These files include graphical information such as geometry and layout (2D/3D) of the decentral conveyor modules. They are transformed from STEP to COLLADA via a special conversion process (Inventor → STEP → FreeCAD → WRL → Blender → DAE).
- Transformation rules (Collada → Collada): These rules describe the semantic lifting of the graphical CDL logic to AutomationML/COLLADA logic. This includes conversion rules between the CDL coordinate system and the COLLADA coordinate system as well as Frame[2] attachment information.

Due to the semantic it is not possible to automate the transformation rules at the moment. Manuel efforts are necessary within the generation process or steps are implemented in a static way. An automation of these steps can follow later on.

6 Export to tools

As already described, the processing of data is done as independent of systems as possible. After the fusion step, data is present in general form which must be exported to the proprietary form of the dedicated end system. To standardize the export process as much as possible, the PCFF conversion tool 'PCFF-View/IO-Assistant' (PCFF configuration with integrated transformator) is equipped with a PlugIn interface. Via this interface, data is passed to an Export-PlugIn which does the transformation to the proprietary format of the dedicated monitoring and control system.

The execution of the provided data within the PlugIns is realized by calling the public method 'Export' parameterized by the file including project information e.g. ‚Layout 1+000+000+100_PR.xml'[3]. This file provides information about additional files to the PlugIn. The PlugIn has to accomplish the following steps:

- Identification of images/views to generate
- Identification of objects (e.g. IO fields, buttons, labels) which shall be generated within the corresponding images
- Identification of information for the filling and nesting of objects within the image (e.g. IO fields which shall be nested with variables or labels/buttons which shall be provided with a special text)
- Identification of variables which must be created (e.g. information to a specific 'address' of an OPC variable of a specific conveyor module)
- Identification of alarms and triggers which must be created

At the moment, PlugIns for the systems ProVis.Visu®, Siemens WinCC, and the Gebhard-visualization are provided. The transformation step from the fusioned data into the visualization system differs dependant on the possibilities and requirements of the system. The PlugIn for WinCC uses the Siemens option package WinCC ODK which provides functionality for the generation of a WinCC project. This export comprises the following steps:

- Generation of a new project with unique name
- Generation of all images/views (hierarchic, group-specific and individual views) of the conveyor system with all controlling elements to display process values, labels, and navigation possibilities, etc.
- Generation of the variables for all IO signals
- Generation of alarm texts to display errors or problems

After the export, the plant is represented in the process control system and can be monitored and controlled during the so called online phase. The communication is done via OPC UA similar as described in [8]. To this end, an own OPC UA server runs on the control box of each conveyor which is directly linked to the control logic and has access to all process data of the conveyor module.

7 Implementation on example plant

Results of an export of the example plants to ProVis.Visu® and WinCC from chapter 3 are depicted in figure 8. It is clearly visible that the graphical representation of the transport plant is quite similar in both systems whereas the control elements (e.g. image changes or exit buttons) are realized system specific.

The processing of the data is done in a standardized manner, but keeps the possibility to react to special features of the end systems. This ensures a seamless integration of the resulting visualization into the end system.

---

[2] Position and orientation in 3D space

[3] The tuple '+000+000+100' is oriented to a 3D vector and describes the view 'Top' on objects.

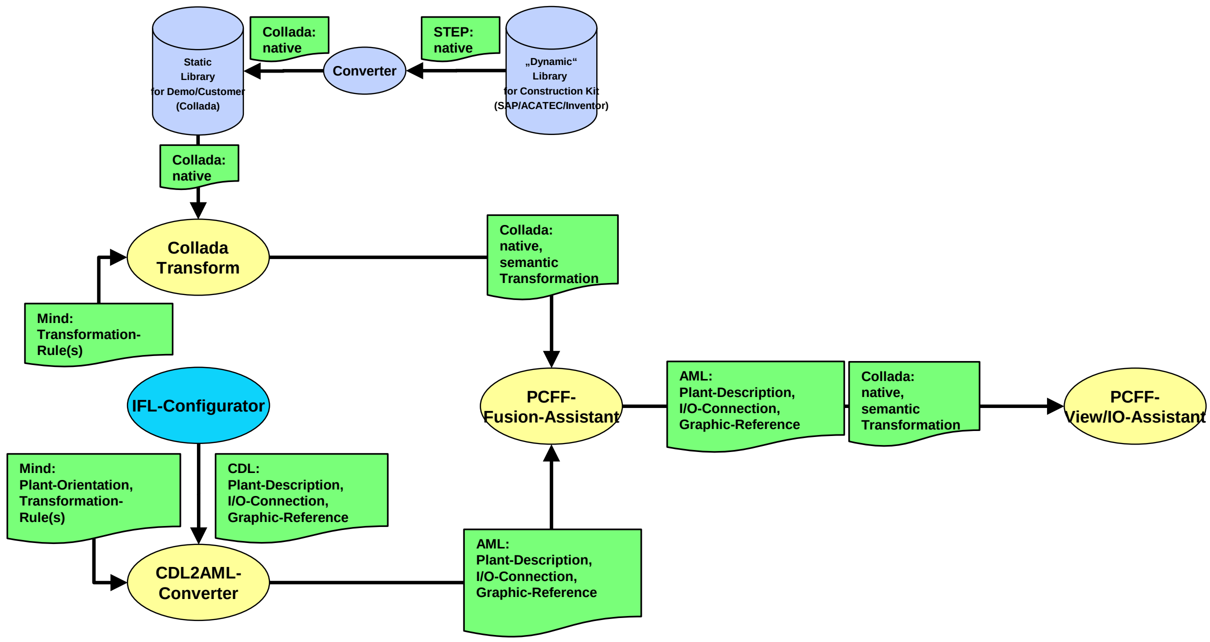

Figure 7. Data fusion

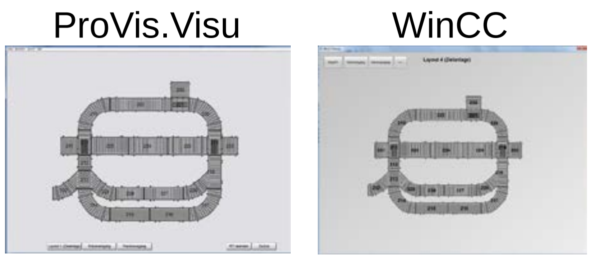

Figure 8. Generated visualization in WinCC and ProVis.Visu®

8 Conclusion and outlook

In the present contribution, an automated framework for the unified generation of monitoring and control visualization in different end systems was presented.

The general concept and framework can be applied to other domains. This is ensured by the usage of standards and the integration to existing (and proprietary) solutions. The claim for standardization is faced by the usage of AutomationML (IEC 62714) [3] to model plant planning data and OPC UA (IEC 62541) [2] as communication and management standard. Further end systems can be easily integrated via the simple PlugIn interface, the connection to the production process via OPC UA also simplifies this extension. This was shown by the fact that it was easy to integrate the proprietary visualization solution of Gebhardt (via the implementation of an export PlugIn) as well which did not possess any API for the external project generation.

References

[1] Die Bundesregierung: Industrie 4.0, http://www.hightech-strategie.de/de/59.php, Stand 30.05.2014.

[2] Wolfgang Mahnke, Stefan-Helmut Leitner, Matthias Damm: OPC Unified Architecture, Springer Verlag, ISBN 3540688986, 2009.

[3] Arndt Lüder, Lorenz Hundt, Nicole Schmidt, Miriam Schleipen: AutomationML-die Architektur. Serie AutomationML Teil 2: Ein Standard für die Verbesserung des Datenaustauschs von Engineering-Werkzeugen. SPS-Magazin Ausgabe 1+2/2013, S. 56-58, 2013.

[4] Siemens: WinCC. http://w3.siemens.com/mcms/human-machine-interface/de/visualisierungssoftware/scada-wincc/seiten/default.aspx, Stand 22.09.2014.

[5] Fraunhofer IOSB: ProVis.Visu®. http://www.iosb.fraunhofer.de/servlet/is/27115/, Stand 22.09.2014.

[6] IEC62541, OPC Unified Architecture (OPC UA), standard series, 2010-2012.

[7] IEC 62714, Engineering data exchange format for use in industrial automation systems engineering - Automation markup language (AutomationML), standard series, 2014.

[8] Robert Wilmes: OPC UA als Kommunikationsbasis in der Industrie 4.0. 23.08.2013, http://www.elektroniknet.de/automation/steuerungstechnik/artikel/100624/, Stand 22.09.2014.