Dear reader,

making engineering collaboration more effective and efficient in a heterogeneous software tool landscape is an ongoing issue for software solution providers and users.

At the upcoming SPS IPC Drives 2015 Electric Automation – Systems and Components, experts from the Christian Doppler research laboratory CDL-Flex at the Vienna University of Technology will present and discuss applications for industry partners based on the AutomationML Hub for the versioned storage and integration of AutomationML data coming from heterogeneous engineering models and tools.

- **AutomationML Hub – Round Trip Engineering** for distributed engineering projects of industrial plants.
- **Simulation Generation** with AutomationML from heterogeneous engineering models.
- **Model-Based Engineering of software tools** for multi-disciplinary engineering.

At the SPS/IPC/Drives 2015 exhibition, scheduled for November 24 - 26 in Nuremberg, Germany, you can find us at the TU Wien booth in hall 4-548, and at the booth of our industry partners logi.cals and LieberLieber in hall 6-331, see details in the section on upcoming events.

In this edition of the newsletter, you will find results from CDL-Flex research and evaluation:

- Lessons learned from Industry Use Cases: **AutomationML Hub – Round Trip Engineering**
- Research Use Case: **AutomationML Analyzer** for the efficient and user-friendly exploration of engineering model data in Automation Systems Engineering
- Inside View: **Model-Based Engineering of software tools** for multi-disciplinary engineering to enable the efficient derivation of software tools.
- New Research Results: **Simulation Generation** with AutomationML from heterogeneous engineering models.
- Consider taking part in the **upcoming events** with experts from the CDL-Flex.

We hope you enjoy the articles and find food for thought on potential improvements and new solutions in your environment. On request, we will be happy to provide you with the cited papers. We are looking forward to discussing your suggestions on issues for research and development to foster alternative solutions for better software data, model, and tool integration in engineering environments.

---

Heinrich Steininginger  
CEO logi.cals  
Heinrich.Steininginger@logicals.com  
www.logicals.com

Prof. Dr. Stefan Biffl  
Head of the Christian Doppler laboratory CDL-Flex at TU Wien  
Stefan.Biffl@tuwien.ac.at  
cdl.ifas.tuwien.ac.at

Peter Lieber  
CEO LieberLieber  
Peter.Lieber@LieberLieber.com  
www.LieberLieber.com
AML HUB - ROUND TRIP ENGINEERING

Software and systems engineering projects require the cooperation of several engineering disciplines, such as electrical, mechanical, and software engineering. However, in engineering tool networks distributed engineering of automated systems often relies on point-to-point data exchange [3] which a) does not sufficiently enable quality and consistency management, b) complicates round-trip engineering, and c) hampers traceability of changes across engineering disciplines.

The need for round-trip engineering arises when the same information is present and relevant in multiple engineering domains and therefore an inconsistency may occur if not all related system elements are consistently updated to reflect a given change. Engineering views on the plant model are not automatically synchronized and changes between engineering operations in cross-discipline context not made visible to the engineers.

![Figure 1: Simple engineering process reflecting the need for Round Trip Engineering](image)

Figure 1 shows a simple engineering process and project role setting. While the plant planner is responsible for defining the overall topology of the automated system, the mechanical engineer, electrical engineer, and PLC programmer are in charge of creating and changing detailed engineering data linked to the plant topology. However, engineering roles would like to define and maintain their discipline-specific topology tree of and their tool-specific view on the automated system. Engineering roles, therefore, should be supported in analyzing the impact of changes on the plant model introduced by others and affecting the engineers’ views. Mechanisms regarding traceability and execution of view-specific checks are required to facilitate the minimization of defects and risks in the overall project planning and to assure overall project quality.

The AutomationML Hub (AML Hub) concept, as shown in Figure 2, systematically integrates tool networks that use the AutomationML standard [4] and enables the automation of engineering processes. The AML Hub reflects contributions of all involved disciplines on a so-called integrated plant model in a structured manner [1]. The AML Hub therefore supports the definition of discipline-specific topology trees and views which may be linked to the integrated plant model for keeping consistency across views.

In order to effectively manage the integrated plant model and engineering role contributions, the AML Hub needs a model description (like in EMF ECore) of the AutomationML schema [2]. Such a description allows, for example, to perform versioning of changes on domain model elements rather than file-format levels.

The AML Hub approach facilitates the efficient versioning of exchanged AML models in tool networks and of operations performed on links between various topology trees and views to improve the traceability of changes across disciplines. Versioning also enables deriving the impact of changes on the integrated plant model and reporting differences to the engineer for improvement of their awareness.

The automation of engineering processes facilitates the synchronization of views on the integrated plant model and the execution of advanced processes such as test automation for quality assurance.

The AML Hub concept has been designed for easy extendibility. For instance,
the plug-in mechanism of the **AML.hub** – an implementation of the concept – allows to insert additional model descriptions and thus systematically enhancing engineering project capabilities. Such descriptions may be elements referenced by the **AutomationML** model, like behavior descriptions in **PLCopen** or geometry models in **COLLADA**.

In a representative standard example, the **AML.hub** was evaluated by the co-

---

**Figure 2:** The AutomationML Hub manages engineering-role specific views on an integrated plant model.

---

**References**


(Richard Mordinyi, Dietmar Winkler)
**RESEARCH USE CASE**

**AutomationML Analyzer**

During the engineering of complex mechatronic systems (e.g., steel mills), a large set of stakeholders, coming from different engineering disciplines, has to collaborate to satisfy project requirements and time frames. Multiple engineering tools from different engineering domains are used and these create diverse, yet somewhat overlapping and interlinked, models of the projected system. Such tools exchange data point-to-point – an approach that makes cross-disciplinary data analysis activities difficult to automate. To ensure optimal project management and to avoid risks of inconsistencies between engineering models created by different disciplines, support is needed for integrating and subsequently analyzing diverse engineering data.

**AutomationML (AML)** is the emerging IEC 62714 standard for facilitating uniform data exchange between engineering tools. AML enjoys an intense industry adoption, especially in relation to the Industrie 4.0 movement. However, even when tool networks use AutomationML, exchanged data may still not be available for querying via a unified interface and cannot easily be linked to support advanced applications that rely on querying project-level data.

The AutomationML Analyzer enables the efficient and user-friendly exploration of engineering model data created in multi-disciplinary engineering settings. Concretely, it enables the integration, browsing, querying, and analysis of diverse engineering models represented in the AML data exchange format.

The AutomationML Analyzer uses Linked Data technology [1] to create an interlinked engineering data space by automatically transforming AutomationML files into Linked Data formats, semantically enriching these and making implicit links across disciplines explicit. This leads to the integration of engineering data from differ-
ent disciplines and tools. Additionally, Linked Data technology provides intuitive an intelligent access gateway to this project-level data space through browsing and navigating interfaces and querying facilities.

For example, Figure 3 depicts an example screenshot of the AutomationML Analyzer, which allows an intuitive Web-based browsing of the engineering information created from the source AutomationML files. The left pane of the interface enables browsing the integrated data along diverse views. Figure 3 shows the “Produktionsmodell” view, which relies on a hierarchical view of the components belonging to the engineered object (see third item from top in the navigation hierarchy). The hierarchy is based on containment relations. In other words, the navigation allows zooming in from the main components of the system to their detailed parts. Other views can also be used for navigation purposes, such as a view based on the main functionalities of the production components.

The right pane of the interface displays all information available about a selected engineering object, in this case a certain Conveyer object. The information displayed here combines information spread across diverse AutomationML files, so the view acts as a merging point for information created by engineers from different disciplines. Particularly interesting are interfaces and relations to other elements, or information specified in different engineering models. Implicit links between the engineering objects in source files are automatically extracted during the transformation from the source AutomationML files and explicitly specified as hyperlinks.

Besides browsing, the AutomationML Analyzer also enables querying the integrated engineering data. Some example queries are: show all composite devices and their sensors; show all interfaces for all sensors; find all devices that exceeded their maximum working hours.

**Benefits.** Domain experts, i.e., engineers, can automatically transform engineering models represented in the AutomationML format into Linked Data – therefore the transition from AutomationML format to the internal format of the AutomationML Analyzer is automated and does not require additional human effort. Engineers can more intuitively browse engineering data from different disciplines by following links made explicit during the transformation into Linked Data. Managers can query data across disciplines to perform cross-disciplinary data analysis activities.

(Marta Sabou, Olga Kovalenko, F. Ekaputra)

**References**

MODEL-DRIVEN ENGINEERING FOR CYBER-PHYSICAL PRODUCTION SYSTEMS

*Cyber-Physical Systems* (CPSs) are combinations of computational elements/processes with physical elements/processes. CPSs involve the use of embedded devices, networks, and software components with feedback loops, where physical processes affect computations and vice versa. CPSs have been used in a wide range of fields including manufacturing [3]. In this context, *Industrie 4.0* [4] is a collective term embracing a number of contemporary automation, data exchange, and manufacturing technologies. The modern production systems engineering projects, envisaged by *Industrie 4.0*, are large and complex, and often involve heterogeneous engineering disciplines (e.g., mechanical, electrical and software engineering) and, thus, heterogeneous artifacts (from software programs to hardware platform specifications and simulation models).

In *Model-Driven Engineering* (MDE), the abstraction power of models is applied to tackle the complexity of systems [2]. MDE follows the principle “everything is a model” for driving the adoption and ensuring the coherence of model-driven techniques in the direction of simplicity, generality, and integration. Historically, MDE has been mainly applied in software engineering [2], but in recent years, the application of MDE has been increasing in the industrial automation domain [7].

We build on the foundations of MDE and existing standards for system modeling (*SysML*), data exchange (*AutomationML*), and simulation (*Functional Mockup Interface*) to provide dedicated support for the engineering process of *Cyber-Physical Production Systems* (CPPSs) realizing the *Industrie 4.0* vision (see Figure 4).

**Figure 4. Engineering of CPS in Industrie 4.0 based on Model-Driven Integration technologies.**

*AutomationML* (AML) [5] is a neutral, free, open, XML-based, and standardized data exchange format for sharing production system (i) structure, (ii) geometry and kinematics, and (iii) logic data. AML has been developed by the *AutomationML* consortium, which consists of companies and academic institutions, including TU Wien.

*SysML* [6] is a graphical modeling language standardized by *Object Management Group* (OMG) for the development of large-scale, complex, and multi-disciplinary systems in a model-based approach. *SysML* derives from the *Unified Modeling Language* (UML) and provides modeling concepts for representing the requirements, structure, and behavior of a system in a coherent system model that...
provides the basis for designing, implementing, and analyzing the system.

*Functional Mock-up Interface* (FMI) [1] is a tool-independent standard to support both model exchange and co-simulation of dynamic models representing physical systems using a combination of XML-files and compiled C-code.

Our research work on MDE for CPPS is currently focusing on:

- The integration of *SysML* and AML models through model transformations and state-of-the-art model-driven technologies;
- Versioning, linking, and co-evolution support for AML models; and
- The integration of discrete and continuous simulation models created with UML/SysML and FMI standards.

The integration of considered standards (SysML, AML, FMI) is conducted in cooperation with and with technology support of our industrial partner *LieberLieber* (http://www.lieberlieber.com)

Further information on our research can be found at www.sysml4industry.org.

---

**References**


(Luca Berardinelli, Emanuel Mätzler, Manuel Wimmer)
SIMULATION GENERATION FROM HETEROGENEOUS ENGINEERING MODELS REPRESENTED IN AUTOMATIONML

Industrial automation systems continue to become more complex and sophisticated. Simulation models play an important role for various scenarios in industrial system design. Simulation models can be used to train and to test human operators, to fine-tune control systems, or for advanced process control.

One of the main issues, which limit the use of simulation models in daily industrial practice, is the time-consuming and error-prone engineering phase, which has been based on manual work by now. The main goal of the presented research is to semi-automate the (re-)design phase of simulation models in order to make the (re-)design phase fast and efficient, as depicted in Figure 5.

Figure 5: High-level workflow of (re-)designing simulation models.

The presented method targets especially large-scale industrial systems as the method assumes that systems consist of pre-defined atomic components (such as pumps, pipes, or tanks). We assume that the functionality of these single components is modeled in a simulation library. The task of the simulation design is thus to select appropriate components from the library, instantiate them, and inter-connect them according to the structure of the real system. To facilitate sharing knowledge about the real system structure, the plant model is imported in the AutomationML data format. The proposed method cannot accept an arbitrary AutomationML file, but rather a plant model or a system model, whose entities are interpretable from the plant topology view point as well as from the view point of signals.

NEW RESEARCH RESULTS

Figure 6: Proposed method for simulation model generation.

The core part of the method, depicted in Figure 6, is the Simulation Generation Interface, whose algorithms are based on the well-proven Bond Graph theory, which was extended by the author to better reflect the needs of latest CAE tools and approaches used by engineering teams.

The Bond Graph method is an engineering approach for describing physical systems and for creating simulation models. Bond Graphs are graphical notations of components, connections, and power flows. To illustrate how a Bond Graph looks like, we selected a hydraulic two-tank system; Figure 7 depicts the piping and instrumentation diagram, adopted from [2].

Figure 7: Two-tank hydraulic model.

The standard use of Bond Graphs means to go through the diagram manually and to extract mathematical equations describing the behavior of the system. On the contrary, the Extended Bond Graphs focus on finding a combination of available components in such a way that the topology of the system is properly modeled. The method supports sig-
nal-oriented simulators, and thus distinguishes between input and output signals of each component. For example, each real pipe can be modeled by one of the following two simulation components:

- **Pipe A** component has pressure loss as input and the component itself calculates the liquid flow according to material and geometrical characteristics of the pipe; while
- **Pipe B** component has liquid flow as input, and calculates the pressure loss implied by this component.

Having such an assumption of signal-oriented components, the creation of a simulation model can be considered as a combinatorial task of finding appropriate combinations of components.

Figure 8 depicts the **Bond Graph** resulting as a solution of the combinatorial task. It includes the following system components: tanks ($T_1$, $T_2$), valves ($V_1$, $V_2$, $V_3$), pipes ($P_1 - P_9$), and a pump ($E_1$). These components are interconnected via 1-junctions and 0-junctions that are abstractions of serial and parallel connections in the physical world. Furthermore, **Bond Graphs** include directed power flows, denoted by semi-arrows. Important issues of **Bond Graphs** are causality assignments that support determining which of the variables effort and flow is the input variable and which is the output variable for each bond.

**Bond Graphs** can support the following engineering tasks:

1. Automated or semi-automated design of simulation models using simulation component libraries;
2. Design of signal interfaces for the definition of co-simulation units; and
3. Support for the specification of a simulation library structure.

The main innovation in the presented approach is the support for the **AutomationML** data format as the input plant model for the generation of simulation models in signal-oriented simulators.

**References**


(Petr Novak)
### Upcoming Events & Additional Information

**SPS/IPC/Drives 2015 Tools Fair**
- November 24-26, 2015
- Nuremberg, Germany
- [www.mesago.de/de/SPS/home.htm](http://www.mesago.de/de/SPS/home.htm)

Experts from CDL-Flex will be present also this year on the *SPS/IPC/Drives* from November 24 to 26 in Nuremberg, Germany. At the booth of the Technische Universität Wien (hall 4-548), you will get insight into the latest results from applied research, which have been worked out in cooperation with industry partners. Also, the CDL-Flex industry partner logi.cals will show development results in hall 6-331. Come with a Free Ticket to the *SPS/IPC/Drives* and visit us for a Viennese Apfelstrudel! For more information, please contact Dietmar Winkler at dietmar.winkler@tuwien.ac.at.

**Software Quality Days 2016 (SWQD)**
- January 18 - 21, 2016
- Vienna, Austria
- [http://software-quality-days.com/](http://software-quality-days.com/)

Experts from CDL-Flex have been organizing the scientific program of one of the largest events on software quality and process improvement, the Software Quality Days in Vienna. “Quality – The Future of Systems- and Software Development” is the motto of the upcoming event with more than 60 presentations, 30 industry demonstrations, and over 300 participants.

### Reader's Picks


### CDL-Flex Results Online

Do not miss the latest presentations, use cases, videos, and screen casts of implemented prototypes on the CDL-Flex Website at [http://cdl.ifstuwien.ac.at](http://cdl.ifstuwien.ac.at)