CDL-Flex Industry Newsletter

October/November 2012

Editorial

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 2012 exhibition for electric automation technology** 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 Automation Service Bus® (ASB) for the integration of software tools, data, and engineering processes:

- **Optimized tool chains** for distributed engineering of industrial plants
- **Coupled simulations** of industrial processes (SCADA & Simulation)

At the SPS/IPC/Drives 2012 exhibition you can find us at the **TU Wien booth** in hall 1-558, see details in the section on upcoming events.

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

- Lessons learned from Industry Use Cases: **Traceable Tool Chain Design Patterns**
- Research Use Case: **Focused Reviews** in Automation Systems Engineering
- Inside View: **Proven Open Source Software** for Integration of Heterogeneous Software
- New Research Results: **Simulation Integration Framework** for Coupled Simulations
- 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 discuss your suggestions on issues for research and development to foster alternative solutions for better software tool integration in engineering environments.

Heinrich Steininger
CEO logi.cals

heinrich.steininger@logicals.com
www.logicals.com

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

Industry Use Case: Traceable Tool Chain Design Patterns

In industrial automation engineering, processes often are represented by tool chains, i.e., result artifacts from one tool/domain are communicated to one or more recipient tools/domain experts to build on these artifacts. While this approach links the engineering process well to the technology used, in heterogeneous software tool landscapes there are several issues to address: **Traceability** of the tool chain segments and overview on the engineering process status. **Translation** between data models of the heterogeneous engineering tools in the chain. **Overview** on engineering objects, their status and properties along the process.

Based on the analysis of tool chain needs in several companies, we found three styles of tool chain needs (see Figure 1), which provide tool chain design patterns.

1. Common Workspace. While domain experts mostly work in a private local tool workspace, there is also a common workspace for the project team to exchange artifacts. All artifacts in the common workspace can be seen and manipulated by all qualified project participants at any time, which allows the project team to enact any process the team sees fit. Experts may analyze changes, resolve conflicts, or vote about changes to be applied. The interaction of experts with the common workspace follow the check-in process pattern [1, 2, 3], similar to a software engineering code repository. This process pattern provides configurable solutions to resolve change conflicts, identify invalid engineering objects, and highlight violated consistency checks. In combination with notification techniques (like mails, tickets, issues) the process is a very flexible approach, which also provides process overview based on tracing engineering activities, exchanged engineering objects, and decision making efforts in the project team.

2. Registered mail. While the common workspace style allows maximal freedom for producers and consumers of engineering artifacts to show, see, and manipulate artifacts, many companies want to ensure stronger control of who can see changed artifacts. In the tool chain design pattern *registered mail*, experts who produced a set of changed artifacts select a set of recipients to send the change set to. Therefore, receiving experts only see artifacts that get sent to them, which also limits the information on sending activities to be stored, e.g., artifact status, recipients. In this kind of tool chains the sending experts retain the full flexibility of when sending which information to whom.

3. Automated batch job. While the first two styles require the explicit participation of engineers in the process, the *automated batch job* style aims at minimizing the necessary expert interaction for complex long-running tool chains. A process expert administers the tool chain by configuring and starting a batch job of related tool chains that is to run without human intervention for an extended period. The process may force engineering tools to check in the latest plant data into the common workspace, run integration tests, or create result and error reports.

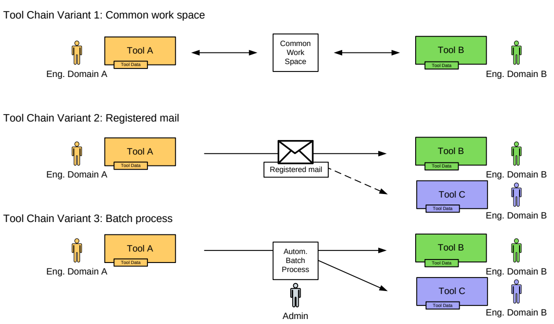

Fig. 1: Tool Chain Variants from 3 companies.

The analysis has also shown that in all companies there is the common need for automated explicit data transformations between local and common workspaces

[4], versioning of exchanged data to notice changes and highlight conflicts [4], and logging engineering activities in order to be capable of reporting, monitoring, and evaluating actions of engineers in specific time frames to identify hot spot in the engineering process [5].

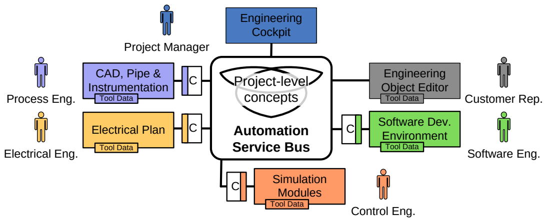

Fig. 2: ASB data and tool integration.

The ASB (see Fig. 2) provides contributions to support all of the tool chain needs found in the surveyed companies: The Engineering Knowledge Base [4] contains various data models about the tools in the chain and enables the definition of model mappings for data transformation. The Engineering Database [4] is a schemaless storage component that stores versions of exchanged data, and supports queries to engineering objects in common workspaces. The workflow engine [6] allows process experts to define and to run engineering processes like the tool chain design patterns, configured according to project requirements.

Key benefits of an ASB solution are:

Simple and fast configuration of quality-assured tool chains propagation of changes between heterogeneous software tools simulation (days instead of weeks).

Use of **industry relevant data exchange formats**, such as AutomationML (www.automationml.org).

Simple analysis of engineering processes for managers and engineers based on a central registry of engineering objects, their states, and tool representations.

References

[1] Winkler, D., Moser, T., Mordinyi, R., Sunindyo, W.D., and Biffl, S. 2011. Engineering Object Change Management Process Observation in Distributed Automation Systems Projects. Paper presented at the 18th European System & Software Process Improvement and Innovation (EuroSPI 2011).

[2] Mordinyi, R., Pacha, A., and Biffl, S. 2011. Quality Assurance for Data from Low-Tech Participants in Distributed Automation Engineering Environments. 2011 IEEE 16th Conference on Emerging Technologies and Factory Automation (ETFA).

[3] Winkler D., Biffl S.: “Improving Quality Assurance in Automation Systems Development Projects”, Book Chapter, Intec Publishing, ISBN 979-953-307-494-7, 2012.

[4] Moser , T., and Biffl, S., Semantic Integration of Software and Systems Engineering Environments; Trans. on Systems, Man, And Cybernetics—Part C: Applications And Reviews, 2011.

[5] Moser, T., Mordinyi, R., Winkler, D., and Biffl, S. 2011. Engineering project management using the Engineering Cockpit: A collaboration platform for project managers and engineers. In Proceedings of the 2011 IEEE 9th International Conference on Industrial Informatics (INDIN'2011)

[6] Biffl, S., Moser, T., Mordinyi, R. 2012. Alle auf “einer” Spur – Automation Service Bus löst Software-Problematik,” Computer & AUTOMATION, no. 6, 2012.

(Richard Mordinyi, 10/2012)

Use Case: Focused Reviews in Automation Systems Engineering

Changes in heterogeneous engineering plans in the automation systems domain require appropriate measures for verification and validation to find defects as soon as they are introduced into the project. Lessons learned from reviews and inspections [2] in software engineering are the foundation for introducing focused reviews in the automation systems domain with tool support. Main goal of this quality assurance approach is to support experts in focusing on critical parts of the system, which have been modified by related stakeholders. In addition, defects, most likely introduced by the change, can be identified more easily and efficiently.

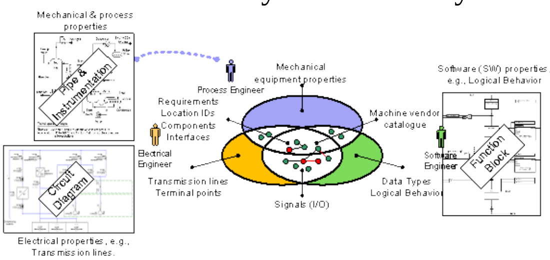

Figure 3: Changes and Defects Across Disciplines and Domain Borders [1].

Figure 3 presents three disciplines and the common concepts between these disciplines. Highlighted engineering objects (marked as red dots in Figure 3) present changes and/or defects, which can be checked by applying focused reviews on common engineering objects.

The semantic integration of data models and comprehensive data exchange [2], enables experts to focus on the most important (i.e., changed and probably defective) system parts[1]. The prototype implementation [3] of the change management process (see Figure 4) provides an early view on changes that are candidates for focused review. First experiences of applying this risk-based approach showed significant advantages from highlighting changes on engineering artifacts, i.e., finding defects earlier and cheaper. Nevertheless, additional benefits are expected to come from providing (a) guidelines to focus experts on important types of defects (e.g., inconsistencies, missing, or wrongly connected engineering artifacts in related disciplines) and (b) in-depth context information to address defects caused by side effects (e.g., change propagation that indirectly affects components).

Figure 4: Highlighted Changes and potential defects at Engineering Object Synchronization.

Thus, ongoing work will focus on tools to (a) provide guidelines to better support experts in conducting focused reviews and (b) building on integrated engineering knowledge from different disciplines to better address most likely risky parts of the system, which may be indirectly affected by a change.

References:

[7] Biffl, S., Moser, T., Winkler, D.: Risk Assessment in Multi-Disciplinary (Software+) Engineering Projects. International Journal of Software Engineering and Knowledge Engineering (IJSEKE), Special Session on Risk Assessment, Volume 21(2), 2011.

[8] Winkler D., Biffl S.: “Improving Quality Assurance in Automation Systems Development Projects”, Book Chapter, Intec Publishing, ISBN 979-953-307-494-7, 2012.

[9] Winkler D., Moser T., Mordinyi R., Sunindyo W.D., Biffl S.: „Engineering Object Change Management Process Observation in Distributed Automation Systems Projects”, 18th European Systems & Software Process Improvement and Innovation Conference (EuroSPI), Roskilde, Denmark, 2011.

(Dietmar Winkler, 10/2012)

[1] For details see the use case description on focused reviews at http://cdl.ifs.tuwien.ac.at/en/download

Inside View: Open Source for Integration of Heterogeneous Software

This section provides a technical inside view on the successful and widely accepted open source software frameworks the **Automation Service Bus** (ASB) uses as solid foundation to provide effective and efficient data, tool, and process integration solutions (see Figure 5).

The basic layer of the ASB architecture is represented by the *Java Virtual Machine* abstracting operating system differences. In order to facilitate modular, extensible, and component-oriented software development, the specification of the *Open Services Gateway initiative* (OSGi) [1] represents the foundation of the ASB architecture. OSGi supports the installation and exchange of software components during run time and enables the integration of those components in a service-oriented architecture (SOA) style. The ASB uses *Apache Felix* as a concrete implementation of the OSGi specification.

On top of *Apache Felix* [2], *Apache Karaf* [3] serves as a software component management tool, backed up by *JLine* to provide a command console for administrators. *Pax-Web* supports an http interface for the ASB using a Web Browser; *Apache Aries* provides Enterprise features like management extensions, persistency, and naming.

Remoting features include *WebService* and *Java Message Service* (JMS) interfaces. *Web Services* (WS) are the de-facto standard for service-oriented architectures and can be easily used in combination with other WS standards. The JMS interface offers the benefit of a simple two-way communication. Client applications can be integrated easier and more transparent than components. The interface uses a message-oriented middleware like *ActiveMQ*, which is also used in *Enterprise Service Bus* implementations.

Apache Wicket [4] in combination with *PaxWicket* and *Blueprint* enable a modular graphical Web User Interface. Further, these frameworks facilitate deployment of web pages as a service which may be dynamically integrated into other solution components.

The basic implementation of the Automation Service Bus, the *Engineering Service Bus* (EngSB) [6], is available as open source software and licensed under the Apache Licence, Version 2.0 [5]. This also implies that licences of any framework the ASB depends on have to be compatible with the Apache Licence. The EngSB project proceeds the steps necessary to become an Apache Project to demonstrate the quality and sustainability of the platform. As a first step the Apache Incubator status has been approached.

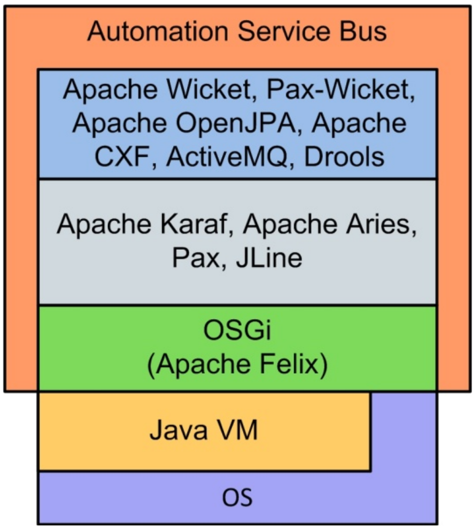

Figure 5: Technology Stack of the Automation Service Bus.

References:

[1] http://www.osgi.org
[2] http://felix.apache.org
[3] http://karaf.apache.org/
[4] http://wicket.apache.org/
[5] http://www.apache.org/licenses/
[6] http://openEngSB.org

(Richard Mordinyi, 10/2012)

New Research Results: Simulation Integration Framework for Coupled Simulations

Simulations of processes usually are a part of supervisory intelligent control systems for design, verification and system control. Simulations and models can be used in a wide range of applications – e.g., operator training, decision support, estimation of unmeasured variables, fault detection, and job planning.

The main goal of the new **simulation integration framework** (SIF) is to provide a configurable integrated environment for simulators and advanced process control tools. Main benefits of the SIF are: **Efficient combination** of simulation modules towards comprehensive simulation (within days instead of weeks). **Simple addition of simulation tools** to the control of the overall simulation. **Data management for traceability** of coupled simulation runs for managers and engineers.

Intelligent control and advanced process control techniques are mostly based on mathematical models of the system behavior. To design and operate these models different kinds of data are needed. Simulation often involves historical and online data and also pre-prepared datasets, testing data, datasets for special scenarios. The simulation integration framework provides unified access to all kinds of data sources needed enabling easy integration.

Usually intelligent control techniques are a composition of different tools like simulators, optimizers, and rule based systems. The goal of the simulation integration framework is to integrate these tools into one environment supporting data exchange in all phases of the simulation life cycle (batch and synchronized tasks, testing and verification etc.).

The final users, e.g., line operators, dispatchers, and operation managers, are usually not experts in the field of control engineering and cannot work with a simulator IDE. Thus it is necessary to connect their native working environment to the simulation framework. A SCADA HMI or web application could be used to control a simulation run. Also the results can be shown in the same applications as well as applied directly to distributed control systems.

We use model-driven tools to simplify the complex configuration of the simulation environment. The architecture of the Simulation Integration Framework (see Fig. 6) consists of two levels: the technical level provides data transformation between tools using Automation Service Bus. On this level the signal routing between the tools, data sources, synchronization etc. have to be configured. Engineering knowledge is captured in a machine-understandable way, to enable flexible and easy to querying and the semi-automated generation of simulation models.

(Radek Sindelar, 10/2012)

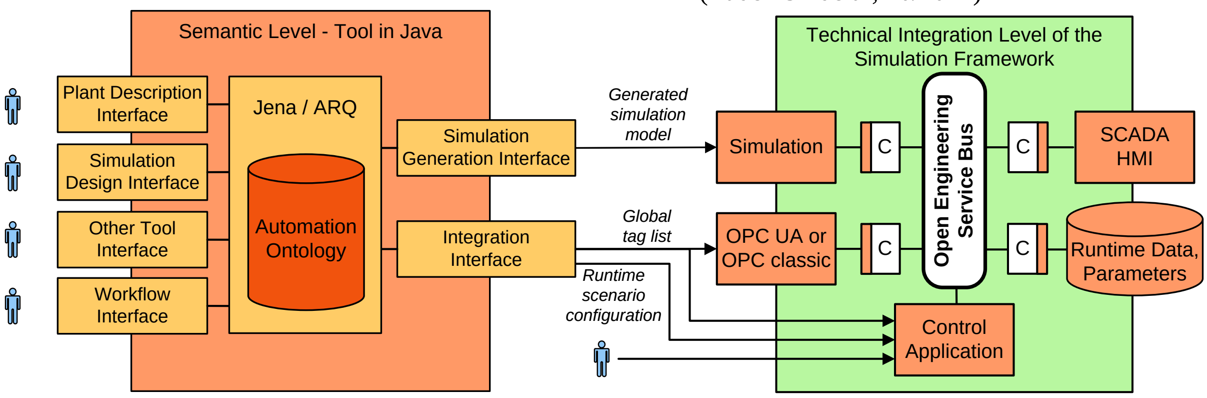

Fig. 6 Architecture of the Simulation Framework: Semantic and technical integration levels.

Recent and Upcoming Events

Automated Software Engineering (ASE) Conference 2012

At ASE 2012 experts of the CDL-Flex showed a tool demonstration of cascading continuous testing in heterogeneous tools, which is relevant for testing several levels of dependent software components across software producing companies.

Emerging Technologies for Factory Automation (ETFA) Conference 2012

CDL-Flex experts co-organized a special session on mechatronic engineering in factory automation and presented their work on data integration for industrial plant engineering in heterogeneous software environments.

Survey on industry integration needs

In order to better understand the needs of practitioners for better software tool integration for the engineering of industrial plants, researchers at the institute for software technology at the Vienna University of Technology conduct a **survey with industrial experts**.

Goals are data collection and analysis on the current situation in industrial practice and needs for better integration of software tools. If you are willing to discuss your view on engineering data and tool integration needs and approaches, please drop us a short mail message to *Stefan.Biffl@tuwien.ac.at*.

Reader's Picks

S. Biffl, R. Mordinyi, and T. Moser, “Anforderungsanalyse für das integrierte Engineering - Mechanismen und Bedarfe aus der Praxis,” *atp edition – Automatisierungstechnische Praxis,* vol. 5, pp. 28-35, 2012.

S. Biffl, T. Moser, and R. Mordinyi, “Automation Service Bus löst Software-Problematik,” *Computer & AUTOMATION*, no. 6, 2012.

SPS/IPC/Drives 2012 Tools Fair, Nuremberg, Nov 27 to 29

Experts from CDL-Flex will be present also this year on the "SPS/IPC/Drives" from November 27 to 29 in Nuremberg. At the booth of the **Technische Universität Wien (hall 1/booth 558)** you 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 **7A-138.**

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 *info@cdl.ifs.tuwien.ac.at*.

Software Quality Days 2013, Vienna, January 15 to 17

Experts from CDL-Flex have been organizing the scientific track of one of the largest events on software quality and process improvement, the Software Quality Days in Vienna. Quality – investment into the future is the motto of the upcoming event with more than 60 presentations, 30 industry demonstrations, and over 300 participants.

For more information, please visit ***http://www.software-quality-days.com***

CDL-Flex Results Online

Do not miss the latest **presentations, use cases, videos, and screen casts** of implemented prototypes on the **CDL-Flex web site**.