Engineering Project Management Using The Engineering Cockpit

A collaboration platform for project managers and engineers

Thomas Moser, Richard Mordinyi, Dietmar Winkler and Stefan Biffl
Christian Doppler Laboratory "Software Engineering Integration for Flexible Automation Systems"
Vienna University of Technology
Vienna, Austria
{*firstname.lastname*}@tuwien.ac.at

Abstract—**Automation Systems Engineering projects typically depend on contributions from several engineering disciplines. While available software tools are strong in supporting each individual engineering discipline, there is very little work on engineering process management and monitoring across multi-discipline engineering projects. In this paper, we present the Engineering Cockpit, a social-network-style collaboration platform for automation system engineering project managers and engineers, which provides a role-specific single entry point for project monitoring, collaboration, and management. We present a prototype implementation of the Engineering Cockpit and discuss its benefits and limitations based on the feedback of our industry partners. Major results are that the Engineering Cockpit increases the team-awareness of engineers and provides project-specific information across engineering discipline boundaries.**

Keywords-process monitoring; multi-domain engineering; (software+) engineering; project dashboard; project management.

I. Introduction

In typical complex Automation Systems Engineering (ASE) projects, such as the engineering of hydro power plants, a range of engineering disciplines, e.g., the mechanical, electrical and software discipline, needs to collaborate within a defined engineering process. While there exist some engineering processes that handle contributions of engineers from different engineering disciplines as a sequence of steps, in practice engineers tend to concurrently update their artifacts, such as documents and plans, originating from different tools in the engineering process, to address new requirements or issues [6].

In multi-disciplinary engineering projects for manufacturing systems or hydro power plants, common concepts, such as signals, bridge the gap between different disciplines on team level. However, the models used in individual engineering disciplines and their best-practice tools often require a range of terms and/or modeling structures to describe a given common concept leading to semantically heterogeneous models [15]. Unfortunately, most engineering tools assume homogeneous data models, where concepts with similar meaning include syntactically similar encoding. Therefore, these tools are hard to seamlessly integrate into a common engineering environment, especially on team level. Nevertheless, efficient data exchange approaches are the foundation for efficient change management processes to increase collaboration between disciplines. Addressing this issue, the major goal of the Engineering Service Bus (EngSB) [4, 5, 7] – a middleware platform for supporting collaboration across tools and disciplines – is to address both technical integration of tools and semantic integration for mapping individual data models across disciplines. The EngSB provides the capability to integrate a mix of engineering tools and backend systems, mobile devices that may go offline, and flexible and efficient configuration of new project environments and engineering processes. In addition technical [4, 5, 7, 9] and semantic integration [3, 10, 11, 17] approaches enable a comprehensive process support and support quality assurance activities across disciplines based on captured process data. In addition, analysis and visualization of process data would enable project monitoring and control and support collaboration and project management on team level.

In this paper, we introduce the *Engineering Cockpit*, a social-network-style collaboration platform for automation system engineering project managers and engineers, applying technical [4, 5] and semantic integration [3, 11, 17] approaches for bridging gaps between heterogeneous ASE project data sources as foundation for comprehensive project monitoring and management. We build on semantic web technology, the Engineering Knowledge Base (EKB) semantic integration framework [10, 11], to explicitly link the data model elements of several heterogeneous ASE project data sources based on their data semantic definitions. We propose the *Engineering Cockpit* as generic framework for project reporting across tool and domain boundaries, and implement a prototype to demonstrate how to calculate a set of metrics for project managers and engineers. We evaluate the feasibility of the *Engineering Cockpit* prototype by performing a set of project-specific queries across engineering discipline boundaries based on real-world ASE project data from our industry partner in the hydro power plant engineering domain. In addition, we discuss the benefits and limitations of the *Engineering Cockpit* based on the initial feedback of our industry partners. Major results are that the Engineering Cockpit increases the team-awareness of engineers and provides the possibility to define project-specific queries across engineering discipline boundaries.

The remainder of this paper is structured as follows: Section 2 summarizes related work on ASE, project management and project cockpits/dashboards. Section 3 identifies the research issues, while section 4 presents the use case for the prototypic implementation of the *Engineering Cockpit*. Section 5 describes the prototype and section 6 discusses benefits and limitations based on the feedback of our industry partners. Finally, section 7 concludes and presents further work.

II. RELATED WORK

This section summarizes related work on ASE, on project management and project cockpits/dashboards.

A. Automation Systems Engineering

Automation systems (AS), such as complex industrial automation plants for manufacturing [6] or hydro power plants [16], depend on distributed software to control systems behavior. In automation systems engineering (ASE) software engineering depends on specification data and plans from a wide range of other engineering aspects in the overall engineering process, e.g., physical plant design, electrical engineering, or process planning. This expert knowledge is embodied in domain-specific standards, terminologies, processes, models, and software tools. Engineering models (e.g., model-based design and testing [2]) help to construct new systems products and to verify and validate the solutions regarding the requirements, specification, and design models. Traditional systems engineering processes follow a waterfall-like engineering process with late testing approaches [1]. Unfortunately, insufficient attention is paid in the field of ASE to capabilities for quality assurance (QA) of software-relevant artifacts and change management across engineering domains [15], possibly due to technical and semantic gaps in the engineering team. Thus, there is considerably higher effort for testing and repair, if defects get identified late in the engineering process.

Collaboration between heterogeneous disciplines and tools requires common concepts for mapping individual models and activities [6, 11]. Our observation at the hydro power plant systems integrator showed that signals are common concepts in this application domain. A signal can be defined as an object used to transmit or convey information[1]. In this paper we define a signal as a common concept for linking information between disciplines [11, 17]. Thus, signals are not limited to electrical signal in electrical engineering, but also include mechanical interfaces in mechanical engineering and software I/O variables in software engineering. In complex automation systems, we define relationships between different kinds of signals from different engineering fields and use them to collaborate and communicate. Thus, the application field called "Signal engineering" deals with managing signals from different engineering disciplines and is facing some important challenges, e.g., (1) to make signal handling consistent, (2) to integrate signals from heterogeneous data models/tools, and (3) to manage versions of signal changes across engineering disciplines.

Depending on the engineering discipline, i.e., electrical, mechanical, and software engineering, individual engineers modify signals frequently during the project course. Because of a missing link between individual disciplines it is hard to monitor and control the entire engineering process on team level by the project management or to track changes across system boundaries by individual engineers.

[1] http://www.merriam-webster.com

B. Project Management and Project Cockpits/Dashboards

Project monitoring and management is an accompanying key activity of project managers along the project course to (a) keep track of the project progress and (b) to take countermeasures in case of deviations with respect to time, budget, and quality [8]. Thus, measuring and analyzing project data is the foundation for process observation and decision making processes. A main task of a project cockpit is to provide the current project state if required by the project management.

In business IT software development software cockpits [14] (or software project control centers [12]) have been developed to enable a comprehensive view on the project focusing on individual roles, e.g., project and quality managers, and selected data sets, e.g., temporal project data, defect data, and issues. Traditional software cockpits focus on (a) collecting data, (b) interpreting them, and (c) visualizing the data according to requirements derived pre-defined roles [12] to support project monitoring in pre-defined time intervals or on demand. In addition, project information, e.g., project organization or time schedules, can be included in the project cockpit [13].

Based on discussions with our industry partners we identified the need for an engineering cockpit approach in the automation systems domain. Nevertheless, after analyzing current software cockpit applications we observed limitations regarding collaboration across disciplines, team awareness, and interaction which hinder efficient implementation in heterogeneous engineering environments.

- *Collaboration.* In automation systems development multiple engineers from different disciplines applying various tools and data models have to collaborate. Thus, an engineering cockpit is a promising approach to bring together the individual disciplines within one single platform.
- *Team awareness*. Lessons learned from social networks can help to improve communication and team work by integrating valuable features within the engineering cockpit, e.g., messaging, visibility of available team members. Depending on the context and role different information sets can be presented, e.g., project overview presentation for project managers and detailed change request information for engineers in assigned disciplines.
- *Interaction*. Traditional software cockpits are used for information purposes with limited interaction capabilities. As we see the engineering cockpit as single entry point to engineering data and the current project, contextual and role-specific tasks can be used to drive project managers and engineers to related activities, e.g., navigating to a certain message or retrieving detailed information after a change request was displayed in the cockpit view.

III. RESEARCH ISSUES

In ASE projects several engineering disciplines contribute artifacts that get updated concurrently and contain data elements, which need to be synchronized with the content of corresponding data elements in data models of other tools. While a common understanding among engineers in a project on engineering domain concepts can be achieved, the tool data models are often semantically heterogeneous and therefore hard to synchronize automatically.

The questions which can be raised are what the current situation of project reporting and monitoring in the ASE domain is, and how its efficiency and effectiveness can be improved. For answering these questions, project managers need to be able to collect data from different heterogeneous engineering tools and perform a set of metrics for further analysis purposes, like project reporting (e.g., average change request response time), project monitoring (e.g., monitoring the current project status or activities) or decision making. From this challenge we derive the following key research issues.

RI-1: Context-specific queries across domain boundaries. How can project managers define and conduct project- and context-specific (in the use case described in this paper, the term context is related to either project-phase-specific or role-specific information) queries across domain boundaries? What types of queries are necessary to gain additional insights on project information and project status? What potential benefits regarding project management decisions does an integrated project monitoring tool such as the *Engineering Cockpit* imply?

RI-2: Increased team-awareness of project participants. How can the team-awareness (i.e., the information what other engineers of the same engineering team are working on at the moment) of single engineers or project managers be increased by using social network style (e.g., collaboration and communication mechanisms known from Facebook[2] or Twitter[3]) mechanisms and techniques? Is the *Engineering Cockpit* capable of addressing additional questions of single engineers such as “What is the overall status of the project?” or “What is my specific contribution to the project?”?

RI-3: A single tool for multi-tool monitoring and querying. Can a single point for cross domain monitoring and querying like the *Engineering Cockpit* be successfully introduced to a rather traditional domain such as ASE? Is it additionally feasible to use the *Engineering Cockpit* as starting point for triggering other ASE-related use cases, such as e.g., signal check in or signal deletion?

To answer these research issues we gather requirements from interviews with ASE project experts at our industry partner and develop a prototype of the *Engineering Cockpit*. We then perform initial evaluations and data analyses using the data of a hydro power plant ASE project from our industry partner.

IV. USE CASE

This section presents a multi-disciplinary engineering use case that has been retrieved from an industrial partner developing, creating, and maintaining hydro power plants. Depending on the size of the commissioned power plant there are about 40 to 80 thousand signals to be managed and administrated in different tools of different engineering disciplines. Signals consist of structured key value pairs created by different hardware components and represent one of the base artifacts in the course of developing power plants. With respect to the given signals, the life cycle of a power plant is divided into several phases, each of them reflecting the progress in building the system and the states of the signals. Highly simplified, the following steps are retrieved from the experiences of the industrial partner: (1) First of all engineers start with the requirement & specification phase. In this phase the required data is gathered, such as signals for turbines and generators. It results in the number of sensors, signals and types of sensors. (2) From this data the typology of the system can be created. The output of this step is a number of I/O cards and a network typology. (3) In the next step the circuit diagram is designed. It produces the allocation plan for mechanical resources. (4) Finally the hardware design is finished to be assembled. (5) After this step the Programmable Logic Controller (PLC) software is created to map hardware pin to software pin addresses. (6) Finally the system can be rolled out.

Figure 1. Overview of a multi-disciplinary water power plant engineering use case.

The described process refers to a perfect scenario, whereas in general 25% of all signals change due to changing customer requirements at any point in the life cycle of the development of the power plant. However, the later signals are changed, the more effort has to be invested in coordination with other disciplines and thus the more costs are created. Project managers would welcome monitoring tools allowing them to identify hotspots in the different phases of development. The combination of data sources from different disciplines may provide information about e.g., customer behavior due to the number of change request per project phase, difficult and complex areas in construction due to high number of explicit and implicit changes. However, today’s integrated tool suites often consist of a pre-defined set of tools and a homogeneous common data model, which work well in their narrow scope but do not easily extend to other tools in the project outside the tool's scope.

V. THE ENGINEERING COCKPIT

This section describes the idea behind the *Engineering Cockpit* prototype, its main components by referring to its monitoring, interaction and collaboration, and management capabilities.

---

[2] http://www.facebook.com
[3] http://twitter.com

A. General Layout

The current web-based application prototype[4] supports two views for two roles with information: project managers (see Figure 3) and developers (see Figure 2). The emphasis of the layout of the cockpit is in both cases the same.

On the left hand side (see Figure 2, part 1), the cockpit provides the user with role and context specific interaction capabilities, like the definition of milestones or the creation of new issues for feature requests. The center of the *Engineering Cockpit* (see Figure 2, part 2) supports the project manager with monitoring information reflecting the current status of the project and the developer with the latest issues and discussions (see Figure 2, part 4) related to the project and its development. The right hand side (see Figure 2, part 3) offers menu entries which provide direct collaboration and team awareness characteristics of users and general coordination of the project.

Figure 2. Development view of the Engineering Cockpit.

B. Monitoring and Information Retrieval Properties

The purpose of this section of the cockpit is to provide the user with information about the project relevant to make new decisions. The sources of data used for displaying the results of the query are engineering tools used in the project and engineering data used in the engineering environment.

With respect to the use case described in section IV the engineering cockpit supports two different types of queries at the moment. The first monitoring component (see Figure 3) retrieves information from a single data source, i.e. directly queries a specific entry. The figure shows the project manager the number of open, resolved, and closed issues and thus provides the manager with information indicating the status of the project. The result of the query is displayed in two different diagrams. The upper one presents absolute numbers taking into account the types of the issues. The lower one cumulatively presents the amount of the various issues types by also considering a specific time frame.

Figure 3. Management View of Engineering Cockpit.

The second type of monitoring components combines the results of two queries executed on different tools. Figure 4 provides information about signals and their relation to project phases.

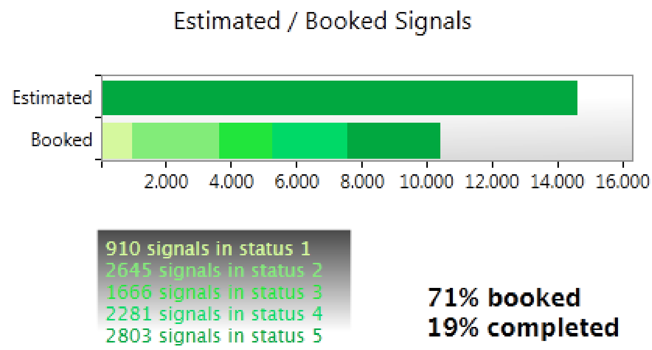

Figure 4. Number of signals and their project phase.

Figure 4 displays an estimated amount of signals typical for a specific project. In relation to that it informs the user about the number of signals currently the project scopes with. The example says that 19% of all signals have reached the final phase (and cannot be changed any more) and that 71% of all estimated signals have already been booked. This also implies that 38% of the estimated signals have not yet been worked with, indicating that the project is probably still in an overall early stage.

Additionally, Figure 5 combines the information about the number of changes with the crafts used in the engineering project and displays it distributed over time. The stacked diagram allows the project manager to find out at which craft most of the changes were triggered, thus it refers to critical areas. As shown in the example, the craft "Turbine" has had to undergo a

[4] The prototype can be accessed online at http://cdl-ifs.tuwien.ac.at/projects/engcockpit

lot of changes. In case the manager wishes to receive more information to that part of the project, it may click on the overview diagram and the *Engineering Cockpit* retrieves information specific to that part only (i.e. "drill down").

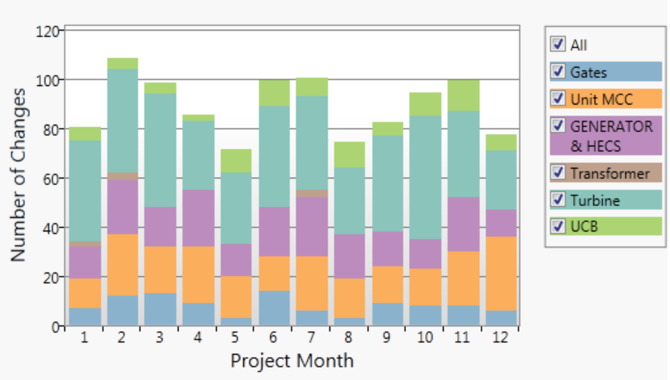

Figure 5. Craft specific stacked signal changes.

C. Interaction Properties

In section IV it has been mentioned that typical projects have to handle 40 to 80 thousand signals. Since in a multi-engineering environment signals may cross the boundaries to other engineering disciplines, no developer is capable of knowing the impacts if he/she changes the properties of a signal. Therefore, the *Engineering Cockpit* explicitly a) facilitates team awareness, b) supports communication between users, and c) enables to set specific actions for coordination purposes. Team awareness (see Figure 2, part 3) is facilitated by offering information about users, their specific role in the project, or about specific project related deadlines and events. Communication (see Figure 2, part 4) is supported by personal chat style interfaces enabling users to directly exchange information and discuss urgent or popular topics.

With respect to coordination and management of the project, a user is capable of using monitored data to explicitly influence the progress of the project. As shown in Figure 2, part 2 the developer may create, change or delete issues from within the *Engineering Cockpit* based on the monitored data, and thus enables context dependent coordination actions. The *Engineering Cockpit* also supports coordination in a way which allows the user to react on monitored data in a role specific manner. Figure 2, part 1 enables developers e.g., to immediately add new signals in case monitored signals or discussions among developers indicate that an important signal has been deleted. It enables the developer to execute project specific engineering tasks.

VI. DISCUSSION

While a common understanding among engineers in ASE projects can be achieved, the tool data models are often semantically heterogeneous and therefore hard to synchronize automatically. This also hinders efficient and effective project reporting and monitoring in the ASE domain, since project managers need to be able to collect data from different heterogeneous engineering tools and perform a set of metrics for further analysis purposes, like project reporting, project monitoring or decision making. From this challenge we derived the following key research issues, which we now discuss.

RI-1: Context-specific queries across domain boundaries. The *Engineering Cockpit* supports two different views for project managers and engineers. Regarding project managers, the *Engineering Cockpit* provides an overview of the project status by summarizing open, resolved and closed issues. In contrast, the *Engineering Cockpit* provides engineers with a collaboration platform showing the current active tasks of other engineers in the same project, and therefore increases the awareness of single engineers regarding potential impacts of their changes, even for engineers working in other engineering disciplines. For the implementation of the prototype for our industry partner, we primarily focused on tool data such as signal lists and issue tracking information, as shown in the previous section. In addition, we used project-specific information such as information on the different project phases and project participants and their organizational units. Using this information effectively allows filtering the displayed data. For visualization, we for the present used simple diagrams types such as bar charts and progress bars, as shown in the screenshots of the *Engineering Cockpit*.

RI-2: Increased team-awareness of project participants. Since the sheer number of signals in typical ASE projects hinders a complete project overview of single engineers or even of project managers, it is very important to provide specific roles only the information that is relevant for them, e.g., in the context of a specific project phase. Therefore, the *Engineering Cockpit* explicitly increases the team awareness of particular roles by offering pre-filtered role-, user- and project-specific information. Furthermore, based on the initial feedback of our industry partner, the *Engineering Cockpit* enables efficient and effective collaboration and communication of project participants by providing state-of-the-art and social-network-style facilities and interfaces such as news feeds, chat panels or personal messages. These communication artifacts are stored for further analyses and extended query capabilities.

RI-3: A single tool for multi-tool monitoring and querying. The *Engineering Cockpit* is capable of a) monitoring current project states and b) retrieving specific aggregated information from multiple data sources. Based on the retrieved and presented results the *Engineering Cockpit* allows the user the execution of role-specific actions, like check-in of new signals. In contrast to specific engineering tools limiting the power of the user to the capabilities of the tool the *Engineering Cockpit* can be seen as an abstract interface enabling access to the functionalities of all engineering tools connected to the Engineering Service Bus. However, the *Engineering Cockpit* can only support access to general tool functions, and thus has difficulties with vendor specific ones. In comparison to multi-purpose tools, like Microsoft Excel, the *Engineering Cockpit* already provides engineering properties which still need to be implemented in the other case.

VII. CONCLUSION AND FURTHER WORK

In typical multi-disciplinary engineering projects for manufacturing systems or hydro power plants, common concepts such as signals bridge the gap between different disciplines on team level. Efficient data exchange approaches such as the Engineering Service Bus (EngSB) [5, 7] provide methods for both technical integration of tools and semantic integration of tool data models for mapping individual data models across disciplines. In addition, analysis and visualization of process data would enable project monitoring and control and support collaboration and project management on team level.

In this paper, we introduced the *Engineering Cockpit*, a social-network-style collaboration platform for automation system engineering project managers and engineers, applying technical and semantic integration approaches for bridging gaps between heterogeneous ASE project data sources as foundation for comprehensive project monitoring and management. We evaluated the feasibility of the *Engineering Cockpit* prototype by performing a set of project-specific queries across engineering discipline boundaries based on real-world ASE project data from our industry partner in the hydro power plant engineering domain.

Major results are that the *Engineering Cockpit* supports two different views for project managers and engineers. The presented prototype displays tool data such as signal lists and issue tracking information, as well as project-specific information such as information on the different project phases and project participants and their organizational units. In addition, the *Engineering Cockpit* explicitly increases the team awareness of particular roles by offering pre-filtered role-, user- and project-specific information; and furthermore offers state-of-the-art and social-network-style means for collaboration and communication of project participants. Finally, the *Engineering Cockpit* allows the user the execution of role-specific actions, like check-in of new signals.

Future work. Based on the discussions with our industry partner, the most relevant features such as the possibility of defining and executing queries will be implemented and evaluated in the context of a real-world engineering project (as described in the use case). Additional implementation is also needed when the potential impact should be propagated to other affected engineers in case of certain changes. The missing part is to establish virtual communication links between groups of engineers by using common signals connecting their crafts.

ACKNOWLEDGMENT

This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria.

REFERENCES

[1] 'GAMP 5 - Good Automated Manufacturing Practice', International Society for Pharmaceutical Engineering (ISPE), 2008

[2] Baker, P., Zhen, R.D., Gabrowksi, J., and Oystein, H.: 'Model-Driven Testing: Using the UML Testing Profile' (Springer, 2007)

[3] Biffl, S., Moser, T., and Winkler, D.: 'Risk Assessment In Multi-Disciplinary (Software+) Engineering Projects', International Journal of Software Engineering and Knowledge Engineering, Special Issue: Software Risk Assessment, 2010, 21, (2), pp. 1-25

[4] Biffl, S., and Schatten, A.: 'A Platform for Service-Oriented Integration of Software Engineering Environments'. Proc. Eight Conference on New Trends in Software Methodologies, Tools and Techniques (SoMeT 09), 2009, pp. 75 - 92

[5] Biffl, S., Schatten, A., and Zoitl, A.: 'Integration of heterogeneous engineering environments for the automation systems lifecycle'. Proc. Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on, 2009, pp. 576-581

[6] Biffl, S., Sunindyo, W.D., and Moser, T.: 'Bridging Semantic Gaps Between Stakeholders in the Production Automation Domain with Ontology Areas'. Proc. 21st International Conference on Software Engineering and Knowledge Engineering, 2009, pp. 233-239

[7] Biffl, S., Winkler, D., Höhn, R., and Wetzel, H.: 'Software process improvement in Europe: potential of the new V-modell XT and research issues', Software Process: Improvement and Practice, 2006, 11, (3), pp. 229-238

[8] Hughes, B.: 'Software Project Management' (Mcgraw-Hill College, 2009)

[9] Kirkham, T., Savio, D., Smit, H., Harrison, R., Monfared, R.P., and Phaithoonbuathong, P.: 'SOA middleware and automation: Services, applications and architectures'. Proc. 6th IEEE International Conference on Industrial Informatics, 2008, pp. 1419-1424

[10] Moser, T.: 'Semantic Integration of Engineering Environments Using an Engineering Knowledge Base', PhD Thesis. Vienna University of Technology, 2009

[11] Moser, T., and Biffl, S.: 'Semantic tool interoperability for engineering manufacturing systems'. Proc. IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2010, pp. 1-8

[12] Münch, J., and Heidrich, J.: 'Software project control centers: concepts and approaches', Journal of Systems and Software, 2004, 70, (1-2), pp. 3-19

[13] Neumann, R., Zbrog, F., and Dumke, R.: 'Cockpit Based Management Architectures', in Abran, A., Braungarten, R., Dumke, R.R., Cuadrado-Gallego, J.J., and Brunekreef, J. (Eds.): 'Software Process and Product Measurement' (Springer Berlin / Heidelberg, 2009), pp. 87-100

[14] Ramler, R., Beer, W., Klammer, C., Wolfmaier, K., and Larndorfer, S.: 'Concept, Implementation and Evaluation of a Web-Based Software Cockpit'. Proc. 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2010, pp. 385-392

[15] Schäfer, W., and Wehrheim, H.: 'The Challenges of Building Advanced Mechatronic Systems'. Proc. 2007 Future of Software Engineering, 2007, pp. 72-84

[16] Schindele, A.: 'logi.CAD Anwendungen in Wasserkraftanlagen', Presentation at logi.cals Powerdays, Monheim, Germany, 2009

[17] Waltersdorfer, F., Moser, T., Zoitl, A., and Biffl, S.: 'Version management and conflict detection across heterogeneous engineering data models'. Proc. 8th IEEE International Conference on Industrial Informatics (INDIN), 2010, pp. 928-935