Extending Mechatronic Objects for Automation Systems Engineering in Heterogeneous Engineering Environments

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

Abstract

Mechatronics is a multidisciplinary field of engineering combining disciplines like mechanical, electronic or software engineering, in order to design and manufacture useful products. Nowadays, mechatronic engineering is well-supported either by using integrated tool suites providing a homogeneous approach to engineering, or by relying on established tool chains consisting of a set of engineering tools connected using a common data exchange format. However, in practice neither tool suites nor tool chains have become a de facto standard in engineering, leading to tedious and often manual integration efforts required to combine specific engineering tools or tool suites.

This paper presents an engineering tool integration framework that allows the definition and usage of mechatronic objects originating from heterogeneous engineering tools, so-called "engineering objects". These engineering objects can additionally include project and organizational information, thus enabling exhaustive engineering process management and monitoring. The presented approach is evaluated in an industrial case study from the hydro power plant engineering domain. Major results are engineering objects that can include heterogeneous data, such as project or organization-specific information, thus enabling automated and therefore more efficient synchronization between the involved engineering disciplines, as well as added-value applications, like project monitoring or quality assured data import and export.

1. Introduction and Motivation

Mechatronics is defined as an integrative discipline utilizing the technologies of mechanics, electronics and information technology to provide enhanced products, processes and systems [1]. It integrates the classical fields of mechanical engineering, electronic engineering and software engineering at the design stage of a product or a system. Increased flexibility, versatility, intelligence level of products, safety, and reliability as well as lower energy consumption and cost are the gains achieved through applying mechatronic concepts to product design [2]. Therefore, mechatronic engineering nowadays is well-supported either by using integrated tool suites (e.g., Siemens Portal[1] or EPLAN Engineering Center[2]) providing a homogeneous approach to automation systems engineering, or by relying on established tool chains consisting of a set of engineering tools connected using a common data exchange format.

However, in practice neither tool suites nor tool chains have become a de facto standard in automation systems engineering. Technically, a different and somewhat overlapping terminology is being used which often hampers common understanding. This can be seen as a result of traditional development where the mechanical part of a system used to be the most complex and difficult one [3]. But in reality, the complexity is slowly shifting to modeling and software engineering issues, requiring a uniform specification of the discrete and continuous parts of advanced mechatronic systems across all involved disciplines. This leads to tedious and often manual integration efforts required to combine specific engineering tools or tool suites, in order to fulfill the requirements of mechatronic engineering.

This paper presents the *Automation Service Bus* (ASB) [4], an Enterprise Service Bus [5] based integration framework for systematically integrating automation systems engineering tools. The ASB addresses the semantic heterogeneity [6] of the engineering tools by modeling and providing the common concepts of the involved engineering disciplines using an explicit and machine-understandable format, the so-called "engineering objects" (EOs). These EOs can additionally include project and organizational information, thus enabling exhaustive support of a set of different engineering processes, as well as automation support for the synchronization between the involved engineering disciplines. Furthermore, they allow for added-value applications using these EOs, such as the Engineering Cockpit [7], an

[1] http://www.automation.siemens.com/mcms/topics/en/tia
[2] www.eplan.de/products/mechatronic/eplan-engineering-center/?L=1

automation systems engineering project monitoring tool with extensive data analysis capabilities, or the Engineering Object Editor [8], an add-in allowing the quality assured import and export of EOs to/from MS Excel.

Based on the challenges and the proposed approach of defining and using EOs in the ASB, the following research issues are discussed in this paper:

- How to define and use mechatronic objects, if the engineering processes use a range of heterogeneous engineering tools and/or tool suites?
- What benefits does the inclusion of project and organizational information into mechatronic objects involve?
- Do these extended mechatronic objects (or short EOs) facilitate existing engineering processes by providing automation support or better quality assurance?

The presented approach is evaluated in an industrial case study based on real-world automation systems engineering project data from a hydro power plant systems integrator. Major results are that the ASB enables the definition and usage of EOs even if the used engineering tools do not use a homogeneous data model, furthermore EOs are not limited to the three major mechatronic engineering disciplines, but can also contain information from other disciplines or domains, such as e.g. project and organizational information. Finally, EOs can support a range of different engineering processes and allow for automated and therefore more efficient synchronization between the involved engineering disciplines.

The remainder of this paper is structured as follows: Section 2 summarizes related work on automation systems engineering, mechatronic engineering and semantic integration of heterogeneous data. Section 3 presents the industrial use case. Section 4 pictures the definition and usage of engineering objects, which is discussed with respect to the research issues in Section 5. Finally, Section 6 concludes the paper and presents future work.

2. Related Work

This section summarizes related work on automation systems engineering, mechatronic engineering and semantic integration of heterogeneous data.

2.1. Automation Systems Engineering

Automation systems (AS), such as complex industrial automation plants for manufacturing or power plants [9] 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 engineering aspects in the overall engineering process, e.g., physical plant design, mechanical and electrical engineering artifacts, and process planning. Therefore, the successful development of modern software-based systems, such as industrial automation systems, depends on the effective and efficient cooperation of several engineering disciplines, e.g., mechanical, electrical and software engineering.

Biffl and Schatten proposed a platform called Engineering Service Bus (EngSB) which integrates not only different tools and systems but also different steps in the software development lifecycle [4, 10] – the Automation Service Bus (ASB) is a modified version of the EngSB for the ASE domain. The EngSB addresses requirements such as the capability to integrate a mix of user-centred tools and backend systems and flexible and efficient configuration of new project environments and SE processes. The EngSB platform introduces the concept of tool domains that provide interfaces for solving a common problem, independent of the specific tool instance used. The abstraction of tool instances by tool domains seems possible since different tools, developed to solve the same problem have, more or less, similar interfaces [10]. On the one hand a tool domain consists of a concrete engineering tool specific interface part used by so called connector components which establish communication connections between a specific engineering tool and the tool domain it belongs to. On the other hand it consists of a general engineering tool independent interface part which enables communication to any other tool domain. This concept allows the EngSB to interact with a tool domain without knowing which specific tool instances are actually present. In heterogeneous ASE environments this capability enables flexible engineering process automation and advanced quality management.

2.2. Mechatronic Engineering

In control system engineering a mechatronic unit is understood as combination of mechanical, electrical, and control related components. This combination is made with the special purpose to ensure a dedicated unit behavior which can be provided to an overall system. Thereby, mechatronic systems are seen as hierarchy of mechatronic units [11]. At the lowest level of the hierarchy so-called basic blocks establish a material and informational flow of the controlled system, actuators, sensors, and information processing units. At the higher levels the mechatronic units connect lower level mechatronic units with information processing units of the higher level [12].

Within the mechatronic engineering process, mechatronic units [13] are considered as units providing dedicated automation system functionalities to the automation system. These functionalities and the conditions required for its provision are the main information modeling paradigm considered within the mechatronic engineering process. Hence, and also following the history of engineering with different independent engineering activities and disciplines, the mechatronic unit is seen within the mechatronic engineering process as consistent combination of information sets of different engineering disciplines. The involved information can be classified either according to the related engineering discipline, to the plant structure, or to the data structures described.

2.3. Semantic Integration of Heterogeneous Data

Semantic integration is defined as the solving of problems originating from the intent to share data across disparate and semantically heterogeneous data [14]. The fundamental reason that makes semantic heterogeneity so hard to address is the independent origin of data sets using varying structures to represent the same or overlapping concepts [15]. From a practical perspective, one of the reasons why schema heterogeneity is difficult and time consuming is that the solution requires both domain and technical expertise: a domain expert who understands the domain meaning of all schemata.

Therani [16] identified the need for a consistent viewpoint and common language for scalable process management. He proposes a task-based ontological framework for integration of information from disparate information sources in multiple domains. Bi *et al. [17]* propose an ontology-based information integration framework for the information integration of mechatronic system multi-disciplinary design tools. Their integration framework adopts component-based architecture and specifies a so-called Mechatronic System Ontology (MSO). The MSO includes all the major objects as well as the relationships of the objects in the process of mechatronic systems design and component interfaces provide services to exchange information.

Wiesner *et al.* [18] present an ontology-based approach for information integration in chemical process engineering. The core of their approach is an expressive knowledge base, which is based on the formal ontology OntoCAPE. The predefined vocabulary within the applied ontologies serves as a stable conceptual interface to the engineering data. Generally, their work differs from the presented approach in that they are able to build their approach on a field-tested ontology (i.e., OntoCAPE) with its extensive substrate of common-sense engineering design knowledge. While this is possible in the chemical process engineering domain, there does not exist such kinds of formal ontologies for many other engineering domains, requiring different methods that rely on efficient and extensible, or even on-the-fly engineering knowledge elicitation.

A good starting point for semantic integration is both the analysis of the interfaces or export artefacts of the involved engineering tools, as well as the identification and analysis of available standards in the problem domain (e.g., AutomationML for the automation engineering domain). AutomationML (Automation Markup Language) is a neutral data format based on XML for the storage and exchange of plant engineering information, which is provided as an open standard [19]. The goal of AutomationML is to interconnect the heterogeneous tool landscape of modern engineering tools in their different disciplines, e.g., mechanical plant engineering, electrical design and control software programming. In their recent works, Drath and Barth [20] introduced the concept of so-called “Collaboration Objects” based on AutomationML. These collaboration objects are conceptually similar to the presented EOs, as they contain data exchanged between different engineering disciplines, however EOs can be dynamically defined because of the underlying ontology-based data models and can further be used during engineering process definition.

3. Use Case

This section presents a multi-disciplinary engineering use case from an industrial partner of hydro power plants. From discussions with the industry partner it is known that the concept of signals is used for collaboration between heterogeneous engineering tools. In a typical engineering project for developing power plants, there are about 40 to 80 thousands signals to be administered and managed from different tools of different engineering fields. Signals help link individual engineering tool models and thus represent objects used to transmit or convey information. The application field ”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.

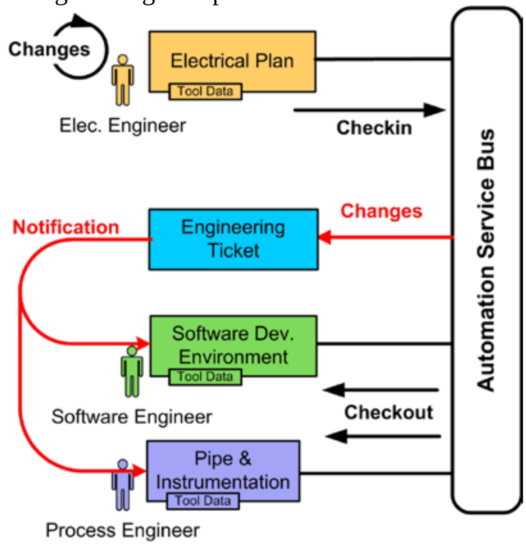

Figure 1: Change management based on mechatronic objects.

Figure 1 shows industry partner relevant engineering tools managing signal information from the tool’s perspective. As shown in the figure, signal information is updated by several tools. Signals are not limited to electrical signal in electrical engineering only, but also include mechanical interfaces in mechanical engineering and software I/O variables in software engineering, and thus may be seen by definition as a mechanical object. The process shown in Figure 1 points out the requirements for knowledge sharing between engineering tools. As in the traditional way of plant engineering where information is passed manually between engineers, the process is performed automatically and is called "check-in". However, as several engineers may work at the same time on the same set of signals it is necessary that each check-in is approved by any other engineer. Whenever the engineer checks-in data the ASB provides information about the type of change to the other engineers by sending an email or creating new issues assigned to them for approval. Automated creation of notifications is necessary in order to minimize surprises in the engineering team due to little communication between project members. Notified engineers receive information about changed signals and review the correctness of the change from their perspective. By accepting the issue the change on a signal is accepted as well.

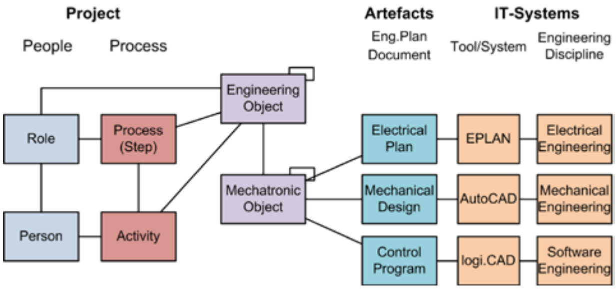

Figure 2: Relations between project, engineering tool, and engineering object models.

Beside engineering discipline related information, signals also store project relevant data (see Figure 2). The progress of a signal also has influence on the state of the overall progress of the project and the system to be built. The life cycle of a power plant is divided into several phases reflected within signals: (1) in the initial phase engineers gather the requirements & specifications for turbines and generators. (2) From this data the typology of the system can be drawn and the circuit diagrams designed. (3) Finally, the hardware design is finished to be assembled. (4) After this step the Programmable Logic Controller (PLC) software is created and tested to map hardware pin to software pin addresses. (5) The system can be rolled out.

The second type of project relevant data refers to the stakeholder who is responsible for triggering changes in the power plant system. External stakeholders, e.g., the customers or the business managers may introduce new requirements or new rules/regulations that affect to the signal changes. Internal stakeholders, e.g., internal engineers or project managers also have their own requirements to change signals in the systems. Storing the reference to the stakeholder of the change allows project managers to bill customers if the number of externally induced changes exceeds limits defined in the contract.

4. Engineering Objects

This section presents the approach using extended mechatronic objects, so-called Engineering Objects (EOs). The following subsections describe the data structure in the ASB framework and engineering process support based on EOs. Finally, two added-value applications based on EOs, the Engineering Cockpit (ECo) and the Engineering Object Editor (EOE) are described.

4.1. Data Structuring in the ASB Framework

The data of two or more ASB tool domains is combined using so-called Engineering Objects (EO) contained in a Virtual Common Data Model (VCDM) [21] The presented concept *signal* is an example of an EO. Figure 3 illustrates the usage mechanism of EOs. In the ASB framework, there exist three data modelling levels, namely the tool instance level (represented by the mechanical planning tool *AutoCAD*, the two electrical planning tools *OPM* and *EPLAN*, the PLC programming tool *logi.CAD*, and *LDAP* for the management of organizational and project-specific data), the tool domain level (represented by the *Mechanical Plan* tool domain, the *Electrical Plan* tool domain, the *PLC Programming* tool domain and the *Organizational Data* tool domain), as well as the newly introduced third layer specifying virtual data model elements, so-called EOs. In the example used in Figure 3, the EO *signal* presents aggregated information of the four tool domains. Since there is no real physical representation of the data aggregated in the EO *signal*, the underlying data model is called VCDM.

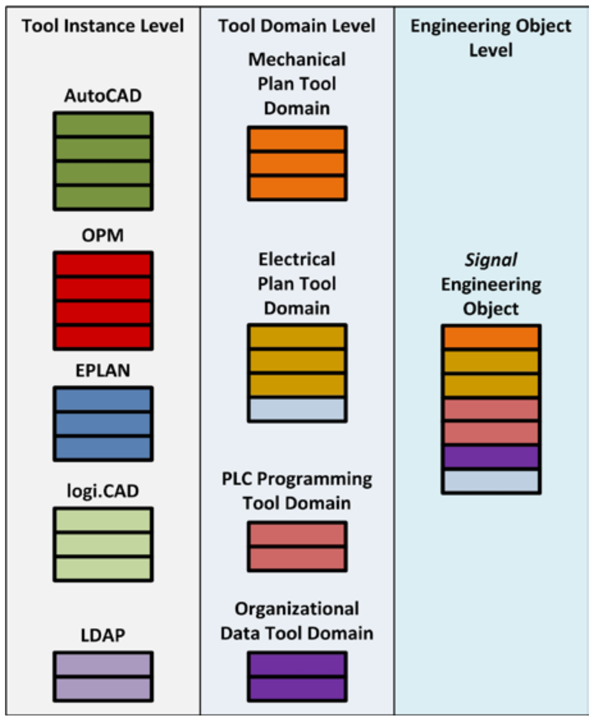

Figure 3. Signals as Example for Engineering Objects in the ASB Framework.

Similar to a database view [22], the information is queried and aggregated on-the-fly, i.e., if some project manager wants to get an overview of all *signals* with a specific signal state. Such tool domain specific information is created during engineering workflow execution and does not have an explicit representation in a particular tool instance. In the ASB framework, such information is called tool domain meta information. This tool domain meta information can be included in an EO-specific view, as shown in the example regarding the signal status.

A typical workflow for querying tool domain data could be defined as follows: (1) the tools can check-in ("push") their data into the ASB framework through so-called tool connectors [4], which on the one-hand side provide the technical connection to the ASB framework and on the other hand side perform two types of conversions. (2) The parsing of proprietary tool data into ASB framework compliant messages and (3) the so-called model transformation, which transforms data satisfying the data models of the tool instances described in the particular tool ontologies into data satisfying the data model described in the explicit tool domain specification in the form of tool domain ontologies which are stored in the Engineering Knowledge Base (EKB) [6]. (4) This transformed data is then stored in the Engineering Data Base (EDB) [21], a versioned database storing data on the level of each tool domain. (5) Queries can now be defined against the data model described in the particular tool domain ontology. (6) these queries then get translated to queries of the data stored in the EDB, (7) and finally the results of these queries are then returned to the user, e.g., in order to be displayed in the Engineering Cockpit (as shown in section 4.3).

4.2. Engineering processes based on EOs

Typical engineering projects (observed at our industry partner) follow a basic sequential engineering process involving several stakeholders from mechanics, electronics and information technology (software). Additional requirements come from project and quality management and other disciplines related to individual phases of the development process. Engineering objects include technical aspects as well as organizational aspects relevant for the project. Figure 4 presents a simplified high-level process for automation systems engineering (ASE) [23]. Individual stakeholders apply different (and isolated) tools and tool-specific data models of engineering objects. Nevertheless, engineers require role-specific views on the engineering objects, e.g., on signals in the hydro power plant application domain.

A sequential process structure makes changes coming from late phases of development (e.g., during test and/or commissioning) challenging, especially if several disciplines/roles are involved. Typically this type of changes (affecting engineers in different disciplines) can lead to project delays if not propagated to related disciplines efficiently. In ASE development projects, where experts work concurrently in distributed in heterogeneous environments, effects of late changes are more critical, more risky, and error prone. Therefore, frequent synchronization between these disciplines is a success-critical issue during development (design and implementation), commissioning, operation, and change request handling.

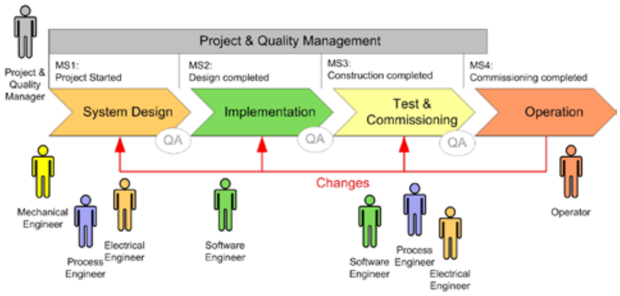

Figure 4. Simplified Sequential Engineering Process.

. For instance, exchanging a sensor type (e.g., changing from analogue to digital devices) might affect mechanical engineers (assembling), electrical engineers (wiring) and software engineers (software variables). Changes of engineering objects have to be propagated efficiently to the affected disciplines to enable concurrent and distributed engineering activities of individual engineers. Figure 5 presents a basic synchronization step of engineering objects applicable in every phase of the sequential workflow. Technical integration of tools and semantic integration of data models support synchronization across disciplines and tool borders.

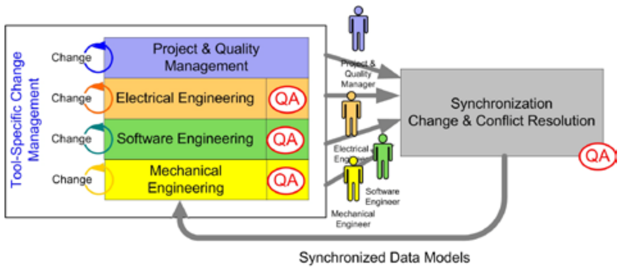

Figure 5. Synchronization of Heterogeneous Engineering Disciplines.

Today we observed a manual synchronization step conducted by experts. Expert knowledge is embodied in engineering objects based on domain specific standards, terminologies, people, processes, methods, models, and tools. Automation supported synchronization based on common concepts enable efficient data exchange in heterogeneous environments leaving discipline-specific aspects within the related disciplines (also relevant for role-specific analysis of engineering processes). Based on the automation supported synchronization approach, engineering objects become observable across disciplines and tool/domain borders.

4.3. Practical application: The Engineering Cockpit

Analyzing, observing, and monitoring the current project state is a key requirement for project and quality managers in ASE projects. Heterogeneity of disciplines and the involvement of various stakeholders in different domains, embodied within engineering objects, makes the observation of the current project state challenging. Today, project data have to be collected, analyzed and visualized manually by experts taking a considerable effort. Thus, the current project state is available infrequently and on request.

Based on the virtual common data model (VCDM) within the ASB Framework engineers and managers are able investigate the current project state based on automated data collection, analysis and presentation of the results based on queries within an Engineering Cockpit [7]. Note that different views on engineering objects enable stakeholder to investigate various aspects of the project, e.g., engineers can focus on tool-specific data and individual open issues and managers can investigate the current project state across tool and domain borders.

Figure 6 presents the implementation of an engineering cockpit prototype to observe engineering objects (i.e., signals), changes, and the impact of changes in an automation system engineering projects. Major components of this cockpit are (1) role specific views on engineering data and activities, (2) various views on engineering objects based on related roles, (3) status data of engineering objects for project progress observation, (4) upcoming context and role-specific tasks, and (5) team awareness for project collaboration support. Figure 6 presents a snapshot of an engineering project at the industry partner using signals as engineering objects. The data presentation section focuses on the project progress, i.e., the number of signals per component over time. Note that different views are based on individual needs of the stakeholders by defining appropriate queries for data analysis and presentation, e.g., number of changes per component, tool, project phase, and over time [7].

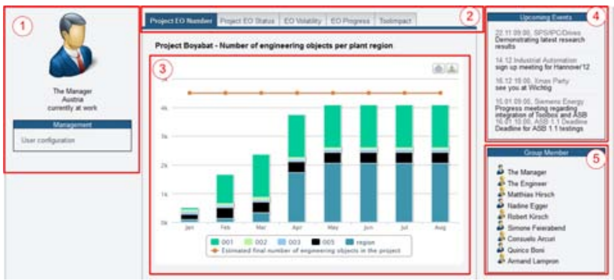

Figure 6. Role-Specific Views on Engineering Objects (Management Perspective).

Because engineering objects include all relevant aspects of the projects, the configuration of additional queries based on common concepts becomes possible to (a) investigate the overall project progress (relevant for project and quality managers) and address individual needs of related engineers from heterogeneous disciplines (e.g., focus on electrical, mechanical, and software aspects). Thus, the engineering cockpit could represent a central starting point for project stakeholders from different disciplines for improving collaboration, interaction, and data exchange in heterogeneous ASE environments.

4.4. Practical application: engineering object editor

As participants in modern automation engineering projects typically work distributed and in parallel, there is also the need to efficiently cope with project participants (e.g., customer representatives) who do not work with the specific engineering tool set but provide important data updates essential to the success of the engineering project. As such participants typically do not have access to the project's data storage a systematic and efficient quality assurance for these inputs is required.

In most of the cases the synchronization between project's integrated database and the external partner's data sets is done by means of Excel spreadsheets as it is a well-established tool used to manage and edit data. While the reintegration of those modifications into the database is not an issue at syntactical level, it is still a major challenge when it comes to provide a systematic and efficient quality assurance (e.g., in case of conflicting data sets) for these inputs.

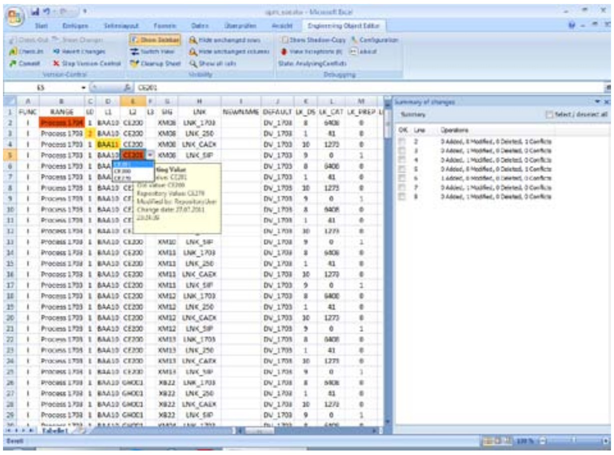

Figure 7. User interface showing changes, conflicting data records, and contact information for contact resolution.

For the retrieval and quality assured storage of engineering objects from the project's database an Excel-add-in called Engineering Object Editor (see Figure 7) has been developed that manages the entire revision control process and states the interface to access the integrated database. Additionally, it helps project members, who stay in touch with external partners to focus on relevant data sets only, by presenting changed or conflicting data sets, hiding negligible information, and providing contact details for efficient conflict resolution.

5. Discussion

This section discusses engineering objects (EOs), a generalized extension to mechatronic objects, as proposed in the Automation Service Bus framework with respect to the research issues defined in the introduction. Engineering objects address integration issues regarding systematically integration of semantically heterogeneous engineering tools for automation systems engineering.

An EO represents a common concept between engineering tools with heterogeneous data models by explicitly encapsulating interrelations between the models and concrete transformations between them. This implies that an EO may also represent a mechatronic object accessing data sources of heterogeneous engineering tools. The trade-off is a higher modeling effort as all data models from all participating engineering tools have to be captured first. Additionally, transformation instructions between the various data models have to be defined. While the first step has to be done manually, the latter may be created semi-automatically [24].

The feasibility of the approach has been demonstrated in two different applications with added value for project managers and engineers. The definition and usage of signals as an EO in a heterogeneous engineering tool environment (see section 3) allows project managers to specify queries on signals and thus to track the progress of a project at any given time. In contrast to the traditional definition of mechatronic objects, EOs propagate changes and thus efficiently perform synchronization steps between heterogeneous data sources. While in previous work this step may have been done manually as well, the proposed approach works fully automated.

Additionally, EOs are not limited to the three major mechatronic engineering disciplines, but can also contain e.g. project and organizational information. In principle any other discipline can be part of an EO. This enables project managers to specify sophisticated queries as relations between models from various disciplines are explicitly captured. The trade-off is once again the higher effort of capturing such information in ontologies. This feature of EOs enable the quality assured import of data from external data sources into the project's data storage as a higher number of plausibility checks may be defined and executed.

Finally, EOs can support a range of different engineering processes and allow for automated and therefore more efficient synchronization between the involved engineering disciplines. This synchronization can be configured in a project-specific way and may be defined by exploiting the connections to other mechatronic objects or other EOs (e.g., the creation of engineering tickets for until now primarily manual engineering process steps).

6. Conclusion and Future Work

Mechatronics is a multidisciplinary field of engineering combining engineering disciplines like mechanical, electronic or software engineering, which is well-supported by either using integrated tool suites providing a homogeneous data model, or by relying on established tool chains using a common homogeneous data exchange format. In industrial practice however, neither tool suites nor tool chains have become a de facto standard, leading to tedious and often manual integration efforts required to combine specific engineering tools or tool suites.

This paper presented the Automation Service Bus (ASB) [4] integration framework that addresses the semantic heterogeneity [6] of the engineering tools by modeling and providing the common concepts of the involved engineering disciplines using an explicit and machine-understandable format, so-called engineering objects (EOs). The presented EO approach was evaluated in an industrial case study based on real-world automation systems engineering project data from a hydro power plant systems integrator, using signals as exemplary EOs.

Major results were that the ASB framework enables the definition and usage of EOs, which are not limited to the three major mechatronic engineering disciplines, but can also contain information from other disciplines or domains, such as e.g. project and organizational information - even if the used engineering tools do not use a homogeneous data model. In addition, EOs support a range of different engineering processes and allow for automated and therefore more efficient synchronization between the involved engineering disciplines. Finally, EOs allow for added-value applications, such as the Engineering Cockpit [7], an automation systems engineering project monitoring tool with extensive data analysis capabilities, or the Engineering Object Editor [8], an add-in allowing the quality assured import and export to/from MS Excel.

Future work will consider user specific configuration of EOs based on individual needs and experiences to better support stakeholders in project execution. Further evaluations in large engineering projects will also be conducted with extended metrics and quantified comparison with related approaches on the market. In addition, usability studies with involved expert roles will be performed in order to ensure the usefulness of the EO approach.

Acknowledgment

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

References

[1] M. Acar, and R. M. Parkin, “Engineering education for mechatronics,” *Industrial Electronics, IEEE Transactions on,* vol. 43, no. 1, pp. 106-112, 1996.

[2] D. A. Bradley, “The what, why and how of mechatronics,” *Engineering Science and Education Journal,* vol. 6, no. 2, pp. 81-88, 1997.

[3] W. Schäfer, and H. Wehrheim, “The Challenges of Building Advanced Mechatronic Systems,” in Future of Software Engineering (FOSE '07), Washington, D.C., 2007, pp. 72-84.

[4] S. Biffl, A. Schatten, and A. Zoitl, “Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle,” in 7th IEEE International Conference on Industrial Informatics (INDIN 2009), 2009, pp. 576 - 581.

[5] C. David, *Enterprise Service Bus - Theory in Practice*: O’Reilly Media, 2004.

[6] T. Moser, and S. Biffl, “Semantic Integration of Software and Systems Engineering Environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, no. 1, pp. 38-50, 2012.

[7] T. Moser, R. Mordinyi, D. Winkler *et al.*, “Engineering Project Management Using The Engineering Cockpit,” in IEEE 9th International Conference on Industrial Informatics (INDIN'2011), Caparica, Lisbon, Portugal, 2011, pp. 579-584.

[8] R. Mordinyi, A. Pacha, and S. Biffl, “Quality Assurance for Data from Low-Tech Participants in Distributed Automation Engineering Environments,” in 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2011), Toulouse, France, 2011, pp. 1-4.

[9] W. Sunindyo, T. Moser, D. Winkler *et al.*, "Workflow Validation Framework in Distributed Engineering Environments," *3rd International Workshop on Information Systems in Distributed Environment (ISDE'11)*, 2011, pp. 1-10.

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

[11] K. Thramboulidis, “Challenges in the Development of Mechatronic Systems: The Mechatronic Component,” in 13th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA’08), 2008, pp. 624-631.

[12] A. Lüder, L. H. M. Foehr, T. Wagner *et al.*, “Manufacturing System Engineering with Mechatronical Units,” in 2010 IEEE International Conference on Emerging Technologies and Factory Automation. (ETFA 2010), 2010, pp. 1-8.

[13] F. Harashima, M. Tomizuka, and T. Fukuda, “Mechatronics - What Is It, Why, and How?,” *IEEE/ASME Transactions on Mechatronics,* vol. 1, pp. 1-4, 1996.

[14] A. Halevy, “Why your data won't mix,” *Queue,* vol. 3, no. 8, pp. 50-58, 2005.

[15] S. Bergamaschi, S. Castano, and M. Vincini, “Semantic integration of semistructured and structured data sources,” *SIGMOD Rec.*, vol. 28, no. 1, pp. 54-59, 1999.

[16] M. Therani, “Ontology development for designing and managing dynamic business process networks,” *Industrial Informatics, IEEE Transactions on,* vol. 3, no. 2, pp. 173-185, 2007.

[17] L. Bi, Z. Jiao, and S. Fan, "Ontology-based information integration framework for mechatronics system multi-disciplinary design." pp. 831 –836.

[18] A. Wiesner, J. Morbach, and W. Marquardt, “Information integration in chemical process engineering based on semantic technologies,” *Computers & Chemical Engineering,* vol. 36, no. 4, pp. 692-708, 2011.

[19] A. Lüder, E. Estévez, L. Hundt *et al.*, “Automatic transformation of logic models within engineering of embedded mechatronical units,” *The International Journal of Advanced Manufacturing Technology,* vol. 54, pp. 1077-1089, 2011.

[20] R. Drath, and M. Barth, “Concept for interoperability between independent engineering tools of heterogeneous disciplines,” in Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on, 2011, pp. 1-8.

[21] F. Waltersdorfer, T. Moser, A. Zoitl *et al.*, “Version Management and Conflict Detection Across Heterogeneous Engineering Data Models,” in 8th IEEE International Conference on Industrial Informatics (INDIN 2010), 2010, pp. 928 - 935.

[22] J. Ullman, “Information integration using logical views,” *Database Theory—ICDT'97*, pp. 19-40, 1997.

[23] D. Winkler, and S. Biffl, "Improving Quality Assurance in Automation Systems Development Projects," *Quality Assurance and Management*: Intec Publishing, 2012.

[24] T. Moser, R. Mordinyi, W. Sunindyo *et al.*, "Semantic Service Matchmaking in the ATM Domain Considering Infrastructure Capability Constraints," *Canadian Semantic Web: Technologies and Applications*, W. Du and F. Ensan, eds., pp. 133-157: Springer US, 2010.