Navigating between Tools in Heterogeneous Automation Systems Engineering Landscapes

Richard Mordinyi, Thomas Moser, 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 the collaboration of several engineering disciplines. While available software tools are supporting individual engineering disciplines quite well, there is very little work on tool collaboration and engineering process automation across discipline boundaries. Following a sequential process structure with distributed and parallel activities, an efficient integration of loosely linked heterogeneous engineering tool models for information retrieval or model consistency checking for quality assurance (QA) requires high human effort. Mechatronic objects provide a logical view on integrated data from mechanical, electrical, and software engineering but there is limited support of heterogeneous and dynamic engineering tool models. Based on so-called engineering objects this paper introduces the capability of single-click navigation between heterogeneous engineering tools taking into account the different views on common engineering concepts. The proposed approach is evaluated by using an example from the process automation domain. Major results are that navigation increases the understanding and traceability of complex relations and thus enables the efficient diagnosis of deviations between models.

Keywords. Automation systems engineering, quality assurance, mechatronical objects, navigation between engineering objects.

I. INTRODUCTION

In typical complex Automation Systems Engineering (ASE) projects, such as the engineering of power plants, a range of engineering disciplines, e.g., mechanical, electrical, and software disciplines need to collaborate within a defined engineering process. Traditional engineering processes in ASE projects typically follow a basic sequential development process, e.g., as provided by water-fall models or V-Model process approaches [1, 2]. While available engineering tools and their data models are strong in supporting individual engineering discipline and process phases, there is very little work on engineering process management, automation, and monitoring across multi-discipline engineering projects - they lack in providing efficient collaboration and interaction between disciplines. Since engineering process activities are usually distributed and parallel, it results in a bleary overview on the real project progress. However, concurrent updates of distributed disciplines need efficient change management - especially in late project phases -, engineering process automation, and effective synchronization mechanisms for quality assurance across discipline boundaries [3, 4]. Otherwise high human effort for (a) synchronizing data of heterogeneous tools, (b) consistency checking for QA, and (c) mapping between related tools and their models is required to provide an accurate overview on project progress [5].

Mechatronic objects aims at providing a logical view on consistent combinations of information sets of different engineering disciplines [6]. They are defined as an integrative discipline utilizing the technologies of mechanics, electronics, and information technology to ensure a dedicated unit behavior which can be provided to an overall system. Mechatronic engineering nowadays is well-supported by (a) using integrated tool suites providing a homogeneous approach to automation systems engineering or (b) relying on established tool chains consisting of a set of engineering tools connected by using common data exchange formats [7, 8]. However, keeping models of different views consistent, especially in the face of model changes, high manual effort [9] is required. Nevertheless, because of this high effort, synchronization is not performed with that high frequency as required by quality assurance personnel. In addition detected defects require efficient mechanism to identify affected components in related engineering artifacts.

This paper presents the so-called navigation approach that connects artifacts of several engineering domains and thus enables to jump between tools in a heterogeneous tool environment with almost none human effort. The capability relies on the Automation Service Bus (ASB) [3] which systematically integrates automation systems engineering tools and addresses the semantic heterogeneity [4] of the engineering tools by modeling and providing common concepts (alias engineering objects) of the involved engineering disciplines. The presented approach is evaluated using well-accepted engineering tools from industry in a use-case from the process automation domain. Major results are that navigation increases the understanding and traceability of complex relations between tool models and that the time needed to identify and find the representation of an engineering object in another software tool is minimized, thus enabling the efficient diagnosis of deviations between the involved models.

The remainder of this paper is structured as the following: section II summarizes related work on automation systems engineering and mechatronic objects, while section III pictures
the industrial use case. Section IV describes the solution concept. Section V presents the evaluation and discusses the evaluation results. Finally, section VI concludes the paper and proposes further research work.

II. RELATED WORK

This section summarizes related work on automation systems engineering and mechatronic objects.

A. Automation Systems Engineering

Automation systems (AS), such as complex industrial automation plants for manufacturing or power plants, rely 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 disciplines. The sequential engineering process depends on the effective and efficient cooperation of several engineering disciplines, e.g., mechanical, electrical, and software engineering.

In heterogeneous engineering environments like in the automation domain, change management and quality assurance are therefore success-critical issues [10]. Different stakeholders coming from various disciplines, e.g., electrical, mechanical, and software engineering domains, apply individual (domain-specific) best practices in a rather sequential engineering process [10]. Observations at industry partners showed (a) a lack of efficient synchronization and data exchange between disciplines, (b) isolated quality assurance activities that focus on the individual domain without considering related disciplines, and (c) limitations on traceability between planning documents in different disciplines. Because of the heterogeneity of engineering and planning documents, e.g., electrical plans, process plans, and software artifacts, synchronization requires a high manual effort by experts.

Biffl and Schatten proposed a platform called Engineering Service Bus (EngSB) integrating different tools and systems and addressing different steps in the systems development lifecycle [3]; the Automation Service Bus (ASB) is a modified version of the EngSB for the ASE domain. The ASB approach enables automation-supported and efficient data exchange in heterogeneous engineering environments[3] to better support change management processes. Based on common concepts [4] defects in overlapping areas become transparent and support engineers in better addressing defects in individual disciplines and across disciplines [10]. Nevertheless, defect removal requires the identification of the defect location in various engineering artifacts and (often across engineering artifacts). Because of different types of artifacts (e.g., tool-data formats and PDF documents) locating engineering objects and defects is still challenging and requires additional effort for searching related items in available documents, even if tools might not be available, e.g., during engineering and commissioning phases. Figure 1 presents (a) the basic sequential engineering process observed at our industry partners and (b) the implemented synchronization process based on the ASB [11]. Figure 1c presents the challenge of linking engineering plans coming from various disciplines as foundation for efficient navigation.

The data of two or more engineering tools connected to the ASB is combined using so-called Engineering Objects (EO) contained in a Virtual Common Data Model (VCDM) [12]; an example for such an EO may be the concept signal. Similar to a database view [13], the information is queried and aggregated on-the-fly, i.e., if project managers want to get an overview of all signals with a specific signal state. For more details on the EO mechanism used to achieve the collaboration between heterogeneous tools, please refer to [14].

B. Mechatronic Objects

Following the history of ASE with different independent engineering activities and disciplines, a mechatronic object can be seen within the overall engineering process as consistent combination of information sets of different engineering disciplines (such as mechanical, electrical, or software engineering) [6]. This combination is made with the special purpose to ensure a dedicated unit behavior which can be provided to an overall system. The involved information can be classified according to (a) the related engineering discipline or (b) the plant structures [15].

For storing hierarchical plant structures, there exist a number of open data formats, such as CAEX (Computer Aided Engineering Exchange). CAEX (IEC 62424) is a neutral data format originally developed in a research project at RWTH Aachen in 2002, which allows the storage of hierarchical object information, e.g. the hierarchical architecture of a plant [16]. Technically, CAEX is based on XML and is defined as XML schema. The original goal of developing CAEX was the industrial lack of a common and established data exchange between process engineering tools and process control engineering tools. The CAEX model-techniques allow the storage of object information that is common across different vendors or tools, e.g., objects, attributes, interfaces, hierarchies, references, libraries, and classes. Therefore, CAEX is mainly used as static data format and not designed for storing dynamic information sets [17].
AutomationML \(^1\) (Automation Markup Language) is an open and neutral data format based on XML for the storage and exchange of plant engineering information \(^1\). The goal of AutomationML is to interconnect heterogeneous tool landscapes of engineering tools in different disciplines, e.g., mechanical engineering, electrical design, HMI development, PLC, or robot control. AutomationML incorporates different standards through links across the used formats: AutomationML links properties and relations of objects in their hierarchical structure implemented with CAEX, geometry and kinematics information implemented with COLLADA \(^2\), and logic (i.e., sequence of actions, internal behavior of objects and I/O connections) implemented with PLCopen XML \(^3\) \(^1\).

III. USE CASE

This section presents a use case from signal engineering that has been retrieved from an industrial partner. It demonstrates a typical process related to the management of signals which are created by different hardware components and represent a base artifact. A signal is a common concept (engineering object) that bridges the gap between different disciplines on team level \(^1\). Depending on the size of the commissioned power plan there are about 40 thousand signals to be managed and administrated in different tools from different disciplines, like:

**E-Plan** \(^4\) (electrical engineering discipline) is a tool to create circuit diagrams, which are used for control cabinets manufacturing. The cable report outputs of EPLAN are used for wiring connections between control cabinets and sensors/consumers.

**logi.CAD** \(^5\) (software engineering discipline) is a hardware independent, multi-function, IEC 61131-3 compatible platform supporting all phases of an engineering project by providing e.g., a graphical editor, debugger, a testing framework, code generator, documentation, or project management features.

Such multi-disciplinary engineering projects typically involve different stakeholders from various disciplines who (a) have to exchange data and (b) interact in case of changes and/or defects. Figure 2 presents the common concepts on team level to link heterogeneous engineering environments, i.e., mechanical, electrical, and software models. Based on the common concepts, defects (red marked items Figure 1) can be identified in overlapping areas during the synchronization process (see Figure 1b). For example, the highlighted defect involves process engineers (P&ID) and software engineers (i.e., function block) who have to discuss and fix the defect.

Typical engineering projects at our industry partner include about 40 thousand engineering objects distributed across

---

\(^1\) http://www.automationml.org

\(^2\) http://collada.org

\(^3\) http://www.plcopen.org/

\(^4\) http://www.eplanusa.com/

\(^5\) http://www.logicals.com
various engineering artifacts. Every engineering artifact can include hundreds of pages (in different tools and/or PDF documents). Thus, locating the defect is quite challenging and can take a considerable effort because the individual identifier of the engineering objects has to be found (a) in the process plan (P&ID) and (b) in the software artifact (function block). After identifying the defect in related artifacts experts start discussing and fixing the defect. In addition we learned from our industry partner that several other document locations have to be checked to avoid side-effects (or the introduction of new defects) during defect repair.

Thus we see the need for supporting search activities of engineers with respect enabling an efficient navigation (i.e., one-click navigation) between engineering artifacts in heterogeneous environments, i.e., mechanical & process plans and software artifacts in the above mentioned example. Therefore, two critical use cases have been identified where efficient navigation can help accelerating the engineering process significantly:

**Use Case 1: Efficient Navigation between Engineering Objects during Engineering Time.** During systems design, i.e., in early phases of systems development (see Figure 1a/b), engineers typically work (a) in parallel and (b) distributed in related disciplines. Therefore, change management and data synchronization are key success criteria for the engineering project. In changes, changes have to be propagated to all related stakeholders (provided by the ASB change management process [10]) as soon as possible to enable collaboration based on a common and stable project base. Supporting engineers in solving conflicts/changes and locating defects by providing efficient navigation mechanisms between artifacts in various disciplines aims at improving engineering processes and reducing searching effort in related artifacts significantly. Note that during engineering phases navigation should enable a direct navigation between engineering tools (which are typically available during design time).

**Use Case 2: Efficient Navigation between Engineering Objects during Systems Commissioning.** The commissioning phase (see Figure 1a/b) is typically located at the plant site where the engineering artifacts and the construction of the plant have been completed. In addition engineering plans have been approved by the customer and are considered as “frozen”, i.e., no changes should be done at this stage. Thus, no tools which enable modifications are available and plans are typically represented in PDF documents. During the commissioning phase engineers search these documents with respect identifying component locations and connection points to see where functional requirements are implemented. Similar arguments apply for the commissioning phase where navigation helps accelerating the search process between artifacts significantly.

### IV. SOLUTION APPROACH

This section describes the ASB supported capability of navigating between tools in a heterogeneous engineering tool environment. Furthermore, this section describes a prototype implementation of an example workflow and presents tool screenshots of the involved engineering tools.

---

![Diagram showing data model used for navigation.](Image)

**Figure 4:** Data Model used for Navigation.

Figure 4 shows the data model used for the navigation use case. From the set of engineering tools as described in the previous section, the three engineering tools which are involved in the engineering process and used for demonstrating navigation use cases are E-Plan Electric, E-Plan PDF (electrical engineering discipline), and logi.CAD (software engineering discipline). E-Plan Electric and E-Plan PDF provide similar functionality, but since not all participating engineers have a full license for using E-Plan electric, E-Plan PDF provides a suitable alternative for viewing engineering plans/artifacts (without providing functionality for data manipulation).

The upper part of Figure 4 presents the three involved tools (left hand side) and the view on the aggregated/joint information provided by the three tools (right hand side). The so-called Signal is defined as an engineering object (EO) in the ASB [3] used for the interaction between three involved tools: the information stored in each Signal instance is filled with information pushed into the ASB by each of the three connected engineering tools. The explicit definition of all data models, i.e., of the data models of all three involved tool as well as of the Signal EO is stored using an ontology-based description in the Engineering Knowledge Base (EKB) component of the ASB [4]. Usually, there exists another layer between the tool data models and the engineering object data models, namely the so-called Tool Domains, which are used for abstracting similar tool data models and functionality. However, since there exists only one tool instance for each used tool domain in the presented example, the data models of tool and tool domain are the same, and therefore the tool domain is omitted in Figure 4 for simplicity reasons. Furthermore, the EKB also holds the mappings between the stored data models, allowing to define which set of one or more attributes of a data model entity of a source model is mapped to another attribute of a data model entity of a target data model. Based on these mappings, transformations between two data models (i.e., tool, tool domain, or engineering object data models) can be defined by the user.

The lower part of Figure 4 shows some exemplary instances of the EO Signal with concrete values defined for each of the attributes described in the models stored in the EKB. In contrast to the models, the concrete individuals are stored in a high-performance database with versioning support, the so-called Engineering Data Base (EDB), which exists for every tool domain connected to the ASB [12].
The table shown in the lower part of Figure 4 displays the aggregated and transformed information of the concrete individuals stored in the particular EDBs of the participating tool domains (or as described in this example, tools), logi.CAD, E-Plan Electric and E-Plan PDF. Therefore, this table can be seen as a direct view of instances of the EO Signal that is filled with data of the mapped tool or tool domain data models. Out of this exemplary data, the different identifier of a Signal in each of the involved engineering tools can be seen, e.g., the Signal in line 2 is identified in logi.CAD using the field SignalKKS with the value GAC03AA001 XG01, in E-Plan the field Eplan with the value 01.GAC03.AA001.XG01 is used, and finally E-Plan PDF uses the field 19007_Placement with the value CMB10/10.3 for identifying this specific signal.

In the following the prototype implementation of the navigation capabilities of the ASB is demonstrated by means of the three engineering tools mentioned before. Figure 5 shows the engineering tool logi.CAD from where the user wants to navigate to either E-Plan or E-Plan PDF. For simplicity reasons it is assumed that each of the tools has already pushed its view of information on a signal into the ASB. Whenever logi.CAD starts up it retrieves every tool that is registered in the ASB and is in relation to a signal in the context of the project open. In order to enable a simple usage, the menu structure in logi.CAD has been extended by an additional entry called “Automation Service Bus@XRef” (see Figure 5 red rectangle). This menu allows the user to select the target tool he wants to navigate to. In case a potentially targeted tool goes offline, the entry representing that tool in the menu is removed as well.

Once the user has clicked on a tool entry in the menu, the ASB identifies the ID of the local representation of a signal in logi.CAD and maps it to the local representation of the target tool. Then the ID of the signal artifact selected in logi.CAD is mapped to the ID of the signal artifact in the target tool. That ID in combination with the navigate command is pushed to the targeted tool which responsibility it is to point out the targeted signal there.

Figure 6 shows the targeted E-Plan tool in case the user selected it in logi.CAD. It has to be noted that in case of E-Plan the tool is not completely integrated with the ASB in the sense of functional integration due to licensing costs. However, E-Plan enables the usage of command parameters to open specific projects and select specific components.

---

6 An aspect of the industrial process plant has been modelled in the Odos-Struger-Laboratory - an industrial automation systems laboratory of the Automation and Control Institute, Vienna University of Technology.
Figure 7 shows the targeted E-Plan PDF in case the user selected it in logi.CAD. The process of navigation is the same as described before.

Since an engineering process may take place concurrently (using three different engineering tools by three different engineers) it is likely that a signal selected for navigation in one tool may have already been deleted in the targeted tool. For that case the ASB supports efficient change management as described in [11]. It propagates changes between various engineering tools according to project specific requirements, like the integration of a ticketing system in order to enforce tractability.

V. EVALUATION AND DISCUSSION

This section discusses the proposed engineering object (EO) with respect to traditional mechatronic objects in the context of navigation between heterogeneous engineering tools. In addition, the benefits for and estimated savings of the industry partner are described briefly.

The major difference between the proposed EO approach and traditional mechatronic objects (e.g., represented using AutomationML) is the nature of the underlying modeling technology. While most mechatronic engineering models rely on meta-modeling technology as underlying modeling technology, the proposed EO approach uses OWL7 ontologies for model representation. As in contrast to static meta-models, ontologies per definition are never seen as complete, this allows an on-the-fly creation of new EO types or updates of existing EOs without the need for adapting the whole ASB framework. In addition, the ASB provides not only data integration (as most of the mechatronic object formats such as AutomationML do), but also integration of engineering tool functionality. Therefore, the in-tool navigation can directly be triggered using the ASB tool connector of the particular engineering tool as described in the previous section. In case of data integration-only tools or frameworks, this navigation functionality has to be additionally implemented, i.e., in the tool responsible for transforming the data from the tool-independent data format to the tool-specific data format.

Efficient navigation can help engineers during the systems design phase (in early engineering phases) and the commissioning phase (in late engineering phases). During systems design (Use Case 1) engineers interact with discipline-specific tools; navigation features support them in (a) identifying defects and (b) finding the location of defects in various engineering artifacts. Thus, related engineers are able to discuss and interact more effective and efficient. In late phases, e.g., during the commissioning phases (Use Case 2), where the engineering project has been completed and the engineering artifacts have been approved, read-only access of engineering artifacts (e.g., using PDF documents) is sufficient and help engineers in finding component locations and connection points more effective and efficient.

Discussions with industry partners showed significant benefits with respect to supporting engineering and commissioning processes: Typical engineering projects include up to 40 thousand engineering objects spread over hundreds of pages (tool-pages or PDF documents) across different disciplines, e.g., mechanics, electrics, and software. Considering an overall search effort of approximately 5 minutes search effort per engineering object following a manual approach, 100 searches (i.e., searching for 0.3% of engineering objects) will lead to 500 minutes (i.e., 1 workday for searching without additional activities). Applying an efficient navigation approach where searching might require less than 30 seconds in a large data base requires 50 minutes. Figure 8 presents the expected effort and benefits of automation supported navigation based on the ASB approach. Note that the x-axis presents the number of navigation activities (# Objects) and the share of objects used for navigation (% Objects) on the base of 40 thousand engineering objects. Note that the expected benefit of applying automation-supported navigation is 90%.

Thus, supporting engineers by providing efficient navigation mechanisms will accelerate engineering processes as well as commissioning phases.

![Figure 8: Expected benefits of automation-supported navigation.](image-url)
VI. CONCLUSION AND FUTURE WORK

Automation Systems Engineering projects typically involve the cooperation of heterogeneous engineering tools from different engineering disciplines. Engineers typically follow a rather sequential process structure with distributed and parallel activities, lacks of efficient synchronization and data exchange between disciplines, and favors isolated quality assurance activities focusing on the individual domains without considering related disciplines. Because of the heterogeneity of engineering and planning artifacts, synchronization requires a time consuming, high manual effort by experts as they have to find, identify, and map corresponding representations of models in different engineering tools in order to ensure plausibility and consistency of those representations.

This paper presented the capability of single-click navigation between heterogeneous engineering tools in the context of the integration framework Automation Service Bus (ASB). Based on common concepts (i.e. engineering objects) the ASB is automatically capable of mapping individual domain-specific data models across engineering disciplines. With the automated resolution of the mapping of common engineering concepts even for heterogeneous representations in software tools the navigation between tools reduces the manual effort of switching between representations and thus increases efficient diagnosis of deviations between models. Future work will consider the integration of several more engineering tools and evaluations in large engineering projects involving multiple organizations.

ACKNOWLEDGMENT

This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. The authors also want to thank their industry partners for their support. We also thank Kristof Meixner, Christoph Prybila, and Michael Steinegger for their valuable contributions.

REFERENCES

[1] "GAMP 5 - Good Automated Manufacturing Practice."