Technical Report

Focused Inspections to Support Defect Detection in Automation Systems Engineering Environments

Dietmar Winkler Stefan Biffl

[1]Christian Doppler Laboratory CDL-Flex, Institute of Software Technology and Interactive Systems, Vienna University of Technology
Favoritenstrasse 9-11/188, AT 1040 Vienna, Austria
<firstname.lastname>@tuwien.ac.at

Technical Report No. IFS-CDL 15-02
Issued: September 2015

Focused Inspections to Support Defect Detection in Automation Systems Engineering Environments

Dietmar Winkler Stefan Biffl

Vienna University of Technology, Institute of Software Technology, Christian Doppler Laboratory “Software Engineering Integration for Flexible Automation Systems” (CDL-Flex)
Favoritenstr. 9/188, A-1040 Vienna, Austria
<firstname.lastname>@ tuwien.ac.at

Abstract. [Context] In Automation Systems Engineering (ASE) Environments, engineers coming from different disciplines, have to collaborate. Individual engineers, e.g., from electrical, mechanical, or software domains, apply domain-specific tools and related data models that hinder efficient collaboration due to limited capabilities for interaction and data exchange on technical and semantic level. Manual activities are required to synchronize planning data from different disciplines and can raise additional risks caused by defects and/or changes that cannot be identified efficiently. **[Objective]** Main objective is to improve (a) engineering processes by providing efficient data exchange mechanism and to support (b) defect detection performance in ASE environments. **[Method]** Software inspection (SI) and software reviews (SR) are commonly used by engineers in Software Engineering by applying well-defined approaches to systematically identify defects early in the development process. In this paper we adapt the traditional SI process for application in ASE environments and provide a software tool to support frequent synchronization and focused reviews. We evaluate and discuss the adapted process in an industry context at a large-scale system integration provider in the hydro power plant domain. **[Results]** Main results were that the adapted process and the software tool can be useful in the application context, i.e., the ASE domain, in order to identify defects early, increase overall product quality, and improve engineering processes in the ASE domain. **[Conclusion]** The proposed adapted inspection approach aligned with the software tool showed promising results to improve engineering projects in the ASE domain.

Keywords: Inspection, Defect Detection, Tool-Support, Automation Systems Engineering Environments, Feasibility Study.

1 Introduction

In automation systems development projects, e.g., developing hydro power plants, manufacturing systems, or steel mills, several engineers have to collaborate and exchange data within the project course [4]. In industry practice engineers work in parallel using individual highly specific engineering tools with heterogeneous and distributed data, e.g., for electrical planning, software planning and construction, process modeling, or simulation. Engineering tools are typically loosely coupled with strong limitations regarding collaboration, data exchange, and defect detection [8][21]. However, frequent data exchange is a pre-condition for engineers to enable related stakeholders and engineers to build on sound technical foundation for project development. The manual synchronization of data is typically executed by experts, who are familiar with at least two related disciplines to synchronize data models and to find defects efficiently [10].

In ASE projects experts have to handle a high number of data sets in large-scale engineering projects; e.g., up to 30k data entities in hydro power plants or several million of data points in steel mill construction projects [22]. Note that the granularity of date entries and data points may vary according to application domain needs. Handing large amounts of data sets require an underlying engineering process that could support synchronization, data exchange, and defect detection. Observations at industry partners showed that engineers follow comparable basic steps within the engineering process. Figure 1 illustrates an observed sequential engineering processes at our industry partner, including defined quality gates (G1-G4) for assessing the quality of project, process, and product attributes.

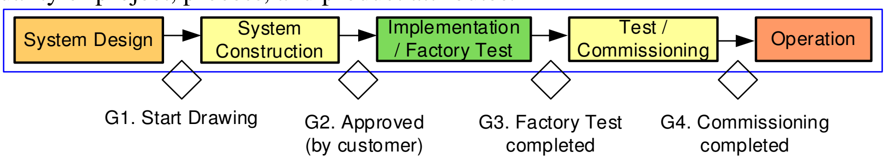

Figure 1: Observed Sequential Engineering Process.

In industrial practice, individual engineers work in parallel (see Figure 2) and apply discipline-specific quality assurance activities on method or tool level. However, the main challenge is to synchronize these planning data from different disciplines to (a) identify defects in overlapping areas early and efficiently, (b) detect inconsistencies between planning data coming from different sources, and (c) support change management processes across disciplines and domain borders.

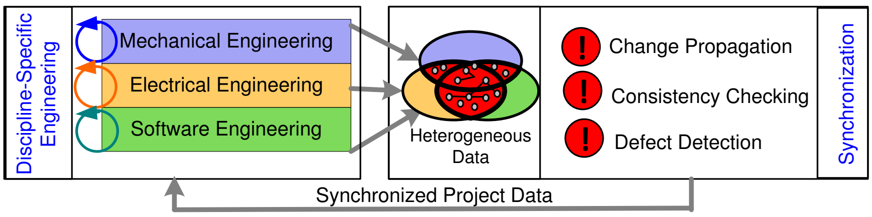

Figure 2: Synchronization Challenges in ASE Projects [10].

Frequent synchronization can help providing a sound technical foundation for all involved disciplines. To overcome technical and semantic gaps between disciplines, there is the need for tool support that enables efficient data exchange and additional quality assurance activities for planning data in the overlapping areas. Figure 2 presents individual quality assurance activities and the concept of manual synchronization between disciplines. During the synchronization process different data sets, e.g., signals or data entities in hydro power plant engineering must be observed and compared to identify deviations that can be real changes, inconsistencies, or candidate defects [10].

In industry practice experts apply several approaches to identify deviations, e.g., by manually comparing lists of data objects or by using automation supported and customized software tools based on spreadsheet solutions (e.g., macros) or data bases. Nevertheless, these solutions can be error prone and can require a high effort for application. Further, additional effort is required for maintaining local (supporting) software solutions in case of local data model changes. Typically, these local expert solutions strongly depend on the availability of the expert who creates and maintains the local solution; thus, this strategy raises an additional risk if the expert is not available any more. To improve (a) defect/change detection and analysis processes and (b) to overcome limitations caused by local expert solutions, we see a strong need for introducing systematic defect detection methods with tool support.

In Software Engineering (SE) systematic review and inspection approaches have been developed and evaluated to support experts or groups of experts (in SE) to efficiently identify deviations and defects systematically. For instance, software inspection is a formal approach for systematic defect detection [11]. Reading techniques [1][9] aim at supporting experts in guiding the inspection (or review) process by providing checklists and guidelines for defect detection. Reading techniques and inspection have been widely investigated in academia (e.g., [1][2][20]) and industry (e.g., [7]). Thus, knowledge and best-practices from software engineering might support experts in the ASE domain to identifying deviations and defects systematically. Based on applied SE best practices, supporting tools can help to focus on the most critical data entities, i.e., the data elements that changed (because of some reason), to increase effectiveness and efficiency of the defect detection approach.

Based on requirements from the ASE domain the main objective is to improve (a) engineering processes by providing efficient data exchange mechanism and (b) to support defect detection performance in ASE environments. We apply best practice software inspection approaches (aligned with the ASE engineering process) and provide a process that supports ASE experts in defect detection in heterogeneous and distributed environments. Further we initially evaluate the adapted process with focused inspections in the application context, i.e., a hydro power plant systems development and integration organization.

The remainder of this paper is structured as follows: Section 2 provides related work on ASE, reviews and inspections, and reading techniques. Section 3 presents the research issues. We describe the solution approach in Section 4. Section 5 presents the prototype implementation and the results of an initial evaluation. Finally, Section 6 discusses the results, concludes and describes future work.

2 Related Work

This section summarizes related work on automation systems engineering environments (2.1) and software inspections (2.2) as well as reading techniques (2.3) as important representatives for early defect detection in context of (software) quality assurance.

2.1 Automation Systems Engineering Environments

Automation Systems Engineering (ASE) Environments include different engineers and stakeholders coming from various disciplines, e.g., electrical engineers, mechanical engineers, and software engineers. Depending on the application domain, additional engineers may represent additional important stakeholders that have to be considered in the engineering process. However, engineers apply different and highly-specific tools and related data models for individual purposes, such as for electrical and mechanical planning or software design [3]. Typically, these tools are not or loosely connected to each other, hinder efficient data exchange (and synchronization) and effective and efficient defect detection [8]; engineering processes become risky and error prone [4].

However, common information sets (i.e., common concepts) are available in ASE projects [13], where experts link heterogeneous data sources (manually or by using individual expert solutions), discuss open issues (at the interface of two or more disciplines), and to identify defects at defined milestones or gates in the engineering process (see central part in Figure 2, "Heterogeneous Data"). Thus, these common concepts represent important glue between heterogeneous and distributed engineering disciplines. For example, in hydro power plant engineering projects, experts use the term "signal" as common concept: signals represent software variables in the software domain or electrical connection points in the electrical domain.

However, to support quality assurance activities in ASE contexts, adapted methods are required to enable effective and efficient defect and inconsistency detection (e.g., [10]). To handle a large number of data entities in automation systems engineering projects (e.g., up to 30k of data entries in hydro power plants or several million of data points in steel mill construction projects), tool support, that is capable of efficiently handling heterogeneous data sets, is needed to support defect and change detection and analysis.

2.2 Reviews and Inspections

Reviews and – more formal – inspections are well-established methods in software engineering to efficiently identify defects and assess the quality of the (software) artifact [1]. Several studies investigated team effects [6], inspection efficiency, inspection effectiveness, and false positives in academia [5], [15]-[20] and industry [7]. Laitenberger *et al.* provide a framework to describe the technical dimension of software inspection and the impact factors along the project course including inspection processes, roles (e.g., moderator, inspectors, readers, and authors), products, and reading techniques [11]. Biffl presents a framework for inspection planning and control [2] on different level, i.e., management plan (project quality planning and reporting), inspection plan (inspection planning and defect detection), and inspection object (defect detection and defect collection). Inspection management is embedded within quality management aspects of the project. See Figure 3 for a detailed view the inspection framework.

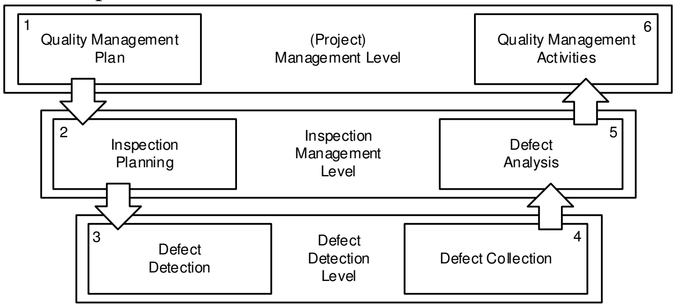

Figure 3: Framework for inspection planning and control [2].

Basically, software inspection follows a defined sequence of steps [11]:

- *Preparation.* Inspection Planning activities include the preparation of material, meeting rooms, and inspection guideline definitions. An (optional) overview enables engineers to get familiar with the project, with artifacts to be inspected, and to get knowledge on the inspection method. This optional step also introduces the project to new team members. Studies showed benefits of inspection for learning effects [19].
- *Defect Detection.* The main goal of software inspection is to identify defects early and in an effective and efficient way. Several reading techniques, i.e., guidelines how to inspect artifacts are available, to support inspectors in defect detection tasks (see Section 2.3. for details).
- *Defect Collection.* Typical inspection processes include (a) individual inspection activities where inspectors aim at identifying defects on their own and (b) team meetings to discuss and aggregate candidate defects towards a team defect lists [6]. Main benefits of team meetings enable discussing individual identified defects, excluding irrelevant (wrongly reported) candidate defects and finding additional defects due to discussion and interaction.
- *Defect Correction and Follow-Up.* Based on aggregated team defect lists, the author can adapt the object under inspection to address raised issues. Depending on the results of the quality assessment, the number of identified defects, and the criticality of the inspection object the moderator or quality managers is responsible for checking whether or not the author has addressed the defects accordingly and/or planning a follow-up inspection (if needed).

2.3 Reading Techniques

The main tasks of reviews/inspection is to analyze an artifact under investigation in order to find defects. Reading technique can support inspectors in finding defects more efficient and effective by guiding inspectors through the inspection process. Several studies investigated different approaches of reading techniques, e.g., [1][14][20] and reported on benefits/weaknesses of different reading technique approaches, e.g.,

- *Ad-Hoc Reading*. Inspectors start reading the document under inspection in a non-systematic way, i.e., without any specific guidelines. There is no clear strategy how to identify defects. This approach (even if frequently used in industry) requires experienced reviewers who identify defects in critical areas (they are aware of based on their knowledge and their experience derived from previous – similar – projects).
- *Checklist-Based Reading* (CBR) techniques represent a more structured strategy to identify defects by stepwise following a pre-defined checklist. Inspectors apply individual checklist items to the artifact and report candidate defects as they are identified. It has been shown that the checklists have to be designed for the application domain and the artifact under investigation as individual aspects might change in different contexts. For instance Thelin *et al.* present a study on a comparison of checklist-based and usage-based reading [17].
- *Usage-Based Reading* and *Perspective-Based Reading* techniques belong to the family of scenario-based reading techniques [1], where the artifact is inspected from different viewpoints, e.g., based on use cases and application scenarios (UBR) or based on individual perspectives (PBR) [16], e.g., from tester, developer, or management perspective. Main advantages include the identification of different types of defects (depending on the individual perspective) and the inclusion of expert knowledge in related disciplines [2][20]. However, some basic guidelines for every perspective are necessary to guide the inspection process efficiently.

Software inspection and reading techniques have a long tradition in software engineering [1] and have been successfully applied in industry. However, in heterogeneous environments (such as in ASE projects) reviews are planned and mainly executed on an ad-hoc basis without systematic support of specific reading techniques. Thus, we see strong benefits of adapting best-practice software inspection approaches to the automation systems domain.

3 Research Issues

Based on the quality assurance needs of engineers in the automation systems domain and the expected benefits of inspection, we derive two main research issues.

RI.1. How can we support quality assurance and defect detection in ASE Environments? This research issue includes two important steps to enable efficient inspections in ASE environments: (a) enable efficient data exchange in environments that include loosely-coupled tools and semantically heterogeneous data models and

(b) apply an inspection process including a well-defined reading technique to the ASE domain to support quality assurance for effective and efficient defect detection. To address this research issue, an approach is required to (a) support efficient synchronization and (b) an adapted inspection and reading technique approach that addresses ASE needs.

RI.2. How can we show the feasibility of the adapted inspection approach in the ASE domain? The second research issue focuses on the evaluation of the defect detection process in an ASE environment in an industry-related application and cost/benefit considerations in selected projects. Thus, we developed a tool-supported change management process approach that is capable of supporting inspection based on real world data derived from our industry partner.

4 Solution Approach

This section presents the solution approach for applying inspection processes in the ASE domain by (a) enabling efficient data exchange in ME environments and (b) the inspection process including a best-practice reading technique that enables efficient defect detection.

4.1 Common Concepts in the Automation Service Bus

In automation systems development projects several disciplines and stakeholders have to interact and exchange data. However, loosely coupled tools and semantically incompatible data models hinder efficient synchronization and include additional risks regarding unidentified defects, inconsistent data models (across disciplines), and limited product quality.

The Engineering Service Bus (EngSB) provides an integration platform to overcome technical heterogeneity of tools and the semantic heterogeneity of data models [3]. To address special needs of the automation systems domain, e.g., in hydro power plant engineering projects, the Automation Service Bus (ASB) has been introduced [22]. Thus, the ASB provides the technical foundation for efficient data exchange in ASE projects.

To overcome semantic heterogeneity of local (and isolated) data models we introduced common concepts [13], i.e., information sets that hold data relevant for all related disciplines, e.g., signals in the hydro power plant domain. Figure 4 illustrates three main disciplines in a typical hydro power plant engineer project, related stakeholders and engineering plans (i.e., data models). Common concepts, described as overlapping areas (e.g., in Figure 2 and in more detail in Figure 4), enable efficient data exchange across disciplines and domain borders, where at least two disciplines have to collaborate. For instance, changes of electrical engineers (e.g., changing a sensor) might affect software engineers to modify related software models. Based on common concepts data can be exchanged between disciplines and engineers can be informed on changes accordingly. By using these semantic technologies, risks of incompatible data models across disciplines and domain borders is minimized [4].

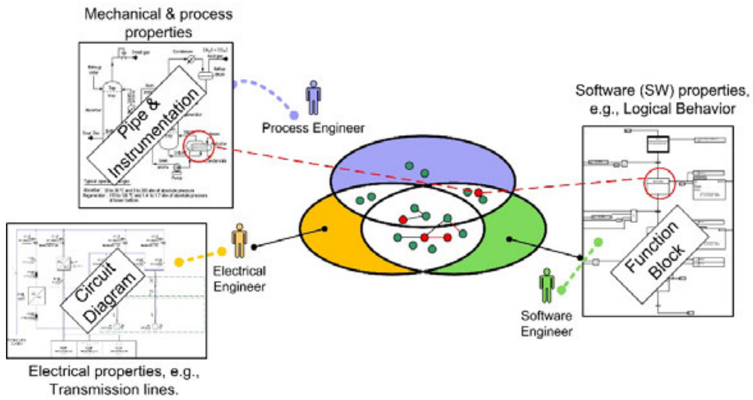

Figure 4: Heterogeneous Data Models and Defects in ME Environments [21].

Beyond efficient data exchange mechanisms between disciplines, common concepts support defect detection by comparing different data models based on these concepts. Deviations can be inconsistencies, defect, or real and required changes. Thus, common concepts represent the starting point for inspecting engineering plans in order to find changes, inconsistencies, and defects.

4.2 Inspection Process in the ASE Domain

Based on semantically integrated data models (coming from various disciplines) an inspection process can be implemented to identify changes, candidate defects and inconsistencies efficiently. Following the traditional SI process (described in Section 2.2 and Figure 5 (lower part, steps 1-6)) the defect detection and collection process can be improved by applying semantic technologies for data integration based on common concepts (Figure 5, steps 1-2, 3' and 5-6).

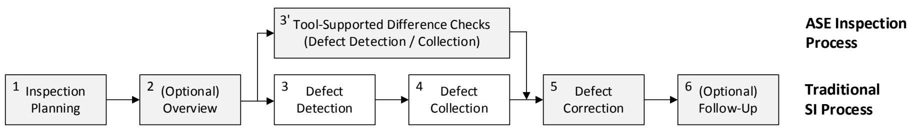

Figure 5: Adapted Inspection Process

In detail, the individual steps of the ASE defect detection approach consists of a set of process steps:

- *Step 1 "Inspection Planning"*. Similar to software inspections the inspection processes have to be scheduled, e.g., by the project and quality manager, according to the project plan or the sequence of quality/decision gates (see Figure 1). Thus, every data element (e.g., a signal) includes an attribute that indicates the current status within the project course; critical gates are G2 "Approved (by the customer)", G3 "Factory test completed", and G4 "Commissioning completed" because changes and/or identified defects require a high effort and cost for repair (in case of available hardware components) or another release cycle by the customer.

- *Step 2 "(Optional) Overview"*. This optional steps aims at introducing the overall project to related candidate inspectors. Similar to the engineering project, the inspection team might be heterogeneous, i.e., including experts from different fields or even non-experts. Thus, the optional phase might be classified as mandatory for ASE projects.
- *Step 3' "Tool-Supported Difference Checks"*. In contrast to traditional SI where step 3 "defect detection" and step 4 "defect collection" are independently executed by individuals and discussed in teams, the ASE inspection approach summarizes both tasks as the data are already available and differences are highlighted (provided by the tool support) [22]. Thus, individual defect detection tasks are obsolete because these tasks are covered by the tool solution and the collected deviations are presented within an ASB dashboard [12]. The inspection team is able to address deviations in a team meeting and process defect classification accordingly. Final result of this step is a set of agreed changes (in case the change was on purpose) and a set of defects that have to be repaired.
- *Step 5 "Defect Correction"*. Similar to traditional software inspections, the authors receive a list of agreed defects to be fixed.
- *Step 6 "(Optional) Follow-Up"*. In traditional software inspection processes the moderator checks the implementation of corrective action and plan a follow-up meeting in case of critical issues (e.g., if defects were not fixed or there is the risk for introducing new defects). In the ASE inspection approach, changes will be integrated again, leading to highlighted deviations that can be checked easily without a high effort. Similar to traditional SI another ASE inspection can be planned if needed.

4.3 Reading Techniques in ASE Projects

Reading techniques support inspectors in efficiently guiding individual inspectors in their review process by applying checklists, use cases and scenarios, or perspectives. Following the heterogeneity of the ASE domain, perspectives seem to be the most promising candidate reading technique for application in ME projects:

- *Electrical Engineers* are responsible for the electrical planning and focuses on electrical issues, e.g., power consumption, wiring, and electrical signal management.
- *Mechanical Engineers* typically focus on the physical setting of the automation system, e.g., by using CAD systems.
- *Software Engineers* are responsible for control applications by using function blocks or structured text approaches.
- *Project and Quality Managers* typically are not interested in individual disciplines but need to keep an overview on the phases, gates, quality, and the project. Thus, they focus on the number of changes/deviations and time constraints within the engineering project.

Depending on the application domain and the phase in the engineering projects, additional engineers and stakeholders might be included. However, all disciplines can use their own highly specific (local) tools and data models and exchange relevant data via the common concept approach. Thus, they can focus on the most relevant issues (within their domain) without considering other disciplines where additional experts are part of the inspection team. The inspection team with different stakeholders (from various disciplines) is comparable to an SI inspection team applying a perspective based reading technique approach, which has been applied successfully in software engineering [16]. In our initial approach we use expert knowledge to decide on a change/candidate defect. However, additional reading technique support can be included, such as defect-driven process approaches [15] that could increase inspection performance (out of scope in this paper).

5 Implementation and Concept Evaluation

To show the feasibility of the adapted ASE inspection approach in context with common concepts and the implementation at our industry partner, we focus on the most critical engineering process at our industry partner, i.e., the change management process in ME environments (signal merging), and present the implementation as foundation for the focused inspection application. Further, we present the capability of having an overview on the current project state from management perspective – an important outcome for the project management and a "byproduct" of the automation supported change management process.

5.1 Change Management and Signal Merge Process

Based on observations and discussions with our industry partner, a large-scale hydro power plant system integrator, we identified the change management process on signal level (i.e., common concepts in the domain), as most critical process. The main challenge is to synchronize different disciplines efficiently.

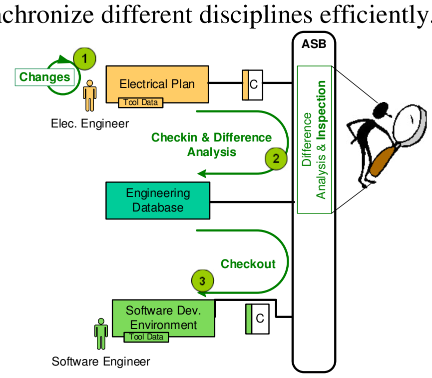

Figure 6: Simplified version of the Signal Merge-Process based on [22].

Figure 6 presents a simplified version of the high-level synchronization and inspection process. Basically, changes (1) executed by a defined discipline (e.g., electrical engineer) are imported by comparing individual signals (on common concept basis) to a central engineering database (EDB) that holds the current version of the overall project (2). The comparison of (at least) two different data models, i.e., the common integrated data model (derived from the EDB) and the modified version of the data model (provided by the electrical engineer), leads to a "merge-view" highlighting the differences between these two data model versions. Note that a deviation can be a real change (required by the electrical engineer), an inconsistency (if changes have not been propagated properly), or a candidate defect that has to be fixed. This view is the foundation for focused inspection where a team of experts inspect the deviations for correctness and decision making (for details see Section 5.2). After completing the inspection process, deviations are accepted (i.e., the new data elements are transferred to the EDB) or rejected (i.e., new data elements are not transferred into the EDB). In parallel, an automated notification mechanism informs engineers of related disciplines on the change/decision discussed during the inspection process. The basic process workflow of the change management process in context of the prototype implementation is also described in [22] and in more detail in [18]. Note that the Automation Service Bus (ASB) [3] represents the technical foundation for the change management process.

5.2 Focused Inspection

Following the adapted SI process approach in the ASE domain (see Figure 5) the focused inspection process includes a set of roles, comparable to the SI process [11]. In our evaluation context the inspection team consists of engineers coming from different disciplines, related to ASE project stakeholders (see Section 4.3), i.e., basic engineering roles (electrical, mechanical, and software engineers) and management roles (systems integrator and the project manager). The adapted inspection approach includes the following steps in context of our evaluation:

- *Inspection Planning.* Based on the project and quality management plan a synchronization (and inspection process) is planned, e.g., prior to iteration completion. At our industry partner these processes are planned on a bi-weekly basis during the *systems design* and the *system construction* phase and on a weekly basis during *implementation* and *commissioning* phases (see Figure 1). Note that different stakeholders participate in different phases. Depending on project management decisions and results of synchronization/inspection processes also the frequency may be adjusted. The inspection artifact is the changed engineering plan/data model (e.g., the electrical plan) represented by an exported "signal-list" describing the system from an electrical perspective. Note that the EDB holds an agreed (and versioned) state of the overall project as reference. Because the inspection team consists of different stakeholders, representing a heterogeneous team, different perspectives help to identify defects, similar to the PBR reading technique approach.
- *Systems Overview.* In case of new project members or very complex inspection

artifacts an overview meeting for the inspection team might be planned. However, in context of our evaluation example, all team members are familiar with the hydro power plant project with years of experience in their domain. Thus, an overview meeting is not required.

- *Tool-Supported Difference Checks.* Following the change management process (see Figure 6), the core part of the ASE inspection approach uses the list of deviations (generated during the merge process) as main input for inspection ("focused inspection"). In our example we have an overall number of 152 signals in the EDB before starting a new synchronization process. The synchronization process showed that 3 new signals have been added, one signal has been changed, and 151 signal have been removed. See Figure 7 for details.

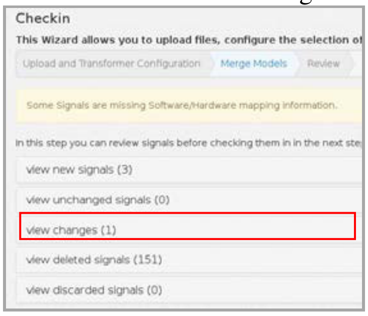

Figure 7: Screenshot: Overview of Signals as basis the Inspection Process.

During the inspection process the inspectors can focus on different categories of changes (i.e., new, changed, or deleted signals) and can accept or reject them. For example the inspectors focus on changes (see Figure 8) and can check modified attributes (see marked area) for correctness. Note that signals attributes refer to a data model in the hydro power plant domain and represents one specific element, i.e., a power connection point. If inspectors agree on a change and neglect a defect, they can select the attribute to be transferred to the EDB, otherwise the old value in the EDB remains valid. In the next step the decisions are reported and related engineers are informed by using a notification mechanism (i.e., Mail).

Figure 8: Focus on Changed Signals during the Merge-Process.

An important benefit of this tool-supported ASE inspection approach is the focus on changes or candidate defects across disciplines and domain borders. As a typical hydro power plant projects includes up to 30k data elements, this approach supports engineers from different perspectives, i.e., engineering roles, in analyzing changes/defects more effective and efficient.

- *Defect Correction.* In the next step, related individual engineers receive the notification on the acceptance/rejection of the change, i.e., in case of a rejection it might be a defect. Based on this notification he can check-out the current and accepted version of the engineering data for his discipline (with focus on data that belongs to his discipline) to continue working on the project.
- *Follow-up.* Based on synchronization (merge) and inspection reports, generated during the notification mechanism, project managers can decide on planning another synchronization cycle. In case of our industry partner no follow-up activities are planned as this approach has been embedded within the overall engineering process.

It is important to note that the inspectors represent individual disciplines/perspectives within the engineering project. This they apply PBR based on their individual expertise. However, introducing systematic guidance for inspectors might increase quality and inspection effectiveness and efficiency. That will remain for future work.

5.3 Management Overview

Beside data exchange and inspection of engineering data on signal level the concept of applying integrated data includes additional benefits related to project monitoring and control. Managers are typically interested in getting stable and on-time information on the project progress. In Moser *et al.* [12] we presented an approach for providing the capability for a continuous observation of engineering data by using an Engineering Cockpit (ECo) solution. Figure 9 presents two prototype evaluation result sets on management level that can be included in the ECo for project monitoring and control.

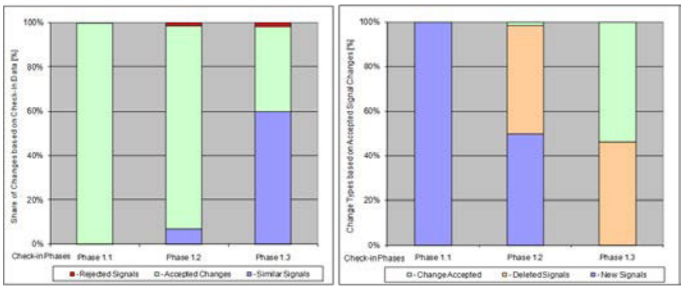

Figure 9: Analysis Results of the Check-in Process from Management Perspective in different Project Phases (Concept Evaluation) [22].

In detail, Figure 9 (left) presents the analysis results in three different check-in phases and the share of accepted signals (real changes) and rejected signals (defects). Figure 9 (right) shows the share of signals related to change types (i.e., new, deleted, and unchanged signals). Based on the implemented change management and signal merge process and the application of the ASE inspection approach, this information can help managers in better assessing the project progress and project quality on management level.

6 Conclusion and Future Work

Frequent synchronization of engineering plans from different disciplines are success-critical issues in ME environments. Loosely coupled tools and data models hinder efficient collaboration and raise additional risks for defects in overlapping areas (Figure 2). In this paper we presented an automation-supported change management process in the ASE domain including an adapted inspection approach that supports engineers and managers in ME environments: (a) efficient synchronization and data exchange based on integrated data and common concepts; (b) efficient defect detection with focused reviews; and (c) capability for timely analysis of project data based on integrated data and common concepts.

The application of the approach at our industry partner, a hydro power plant development and integration organization, showed benefits of integrated data regarding synchronization frequency and data quality. Table 1 summarizes results from discussions with our industry partner on the improvements of ASB application after a year of testing and application in selected real-world contexts. In contrast to previous manual synchronization, the effort for synchronization decreases significantly because of automation-supported data integration. Note that these effort values do not include inspection and discussion activities. We consider a comparable discussion effort (per signal deviation) but significant benefits in preparation and data handling. For instance, in the manual approach experts have to spend effort for identifying the deviation while there is no effort for identifying deviations in the ASB approach. As a consequence the frequency of synchronization increases and the overall project, process, and product quality increases due to defects that can be found early and more efficient in the engineering process.

Table 1: Comparison of Manual and ASB Synchronization Processes.

	Manual	**ASB**	**Change**
Individual Synchronization Effort (without inspection & discussion)	30min	5min	Effort Improvement (-83%)
Avg. Frequency of Synchronization	2 / month	20 / month	Frequency increased: factor 10
Analysis of Data	Days	Seconds	Significant improvement
Inspection and Discussion	-	-	comparable effort

The application of the adapted ASE inspection approach (including tool support) enables defect detection for pre-selected candidate defects/changes and was found useful because it helped the inspection team (which was newly introduced in the engineering process) to better focus on defect detection and discussions rather than focusing on identifying deviations in different engineering plans. Due to tool support (using common concepts and integrated data as driver for inspection) the "defect detection" step and "defect collection" step (step 3 and 4 in Figure 5) can be combined to a "tool-supported difference checks" step 3' (in Figure 5) by using the signal management approach based on common concepts. Individual defect detection (and collection) activities are bundled and supported. Personalized views present the most relevant deviations (candidate defects and/or changes) to the inspection team, i.e., a group of experts, who discuss these deviations and decide whether to accept the deviation as a real change or reject the deviation as a defect. Note that this selection process enables significant improvement of the inspection process because (expensive) experts can focus on the most critical elements, i.e., deviations that could be changes or defects; they do not have to identify candidate changes/defects but see real changes/defects immediately.

Limitations. The nature of automation systems development projects includes a heterogeneous group of stakeholders (i.e., engineers from different disciplines and with different perspectives). Thus, inspection teams can be seen as group of experts applying different perspectives, i.e., perspective based reading techniques. However, in the first application of focused inspection in the ASE domain we did not provide any formal guidelines for different reading technique approaches but we build on the expertise from different disciplines, comparable to perspective-based reading. The test data, used in the evaluation, represent a small snapshot of a large-scale real-world project. We decided to use this small sample to demonstrate the feasibility of the approach. However, a more detailed evaluation including a large set of data will be required. The inspection team was formed from experts at our industry partners, comparable to real-world setting.

Future Work will follow several different research directions: First, to better support different engineers (from different disciplines) guidelines and checklists have to be developed that aim at improving defect detection performance; Second, these different reading technique approaches have to be evaluated in larger industry contexts to identify strength and weaknesses of the reading technique variants; Finally, future work should address approaches and technical solutions of semantic technology to support defect detection and defect detection processes based on common concepts and integrated data.

Acknowledgements

This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth, and the National Foundation for Research, Technology and Development, Austria.

References

[1] Aurum A., Petersson H., Wohlin C.: State-of-the-Art: Software Inspection after 25 years, In: Journal of Software, Testing, Verification and Reliability, 12(3), pp.133-154, 2002.

[2] Biffl S.: Inspection Techniques to support Project and Quality Management, Habilitation, Shaker, ISBN: 3-8265-8512-7, 2001.

[3] Biffl S., Schatten A., Zoitl, A.: Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle, In: Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, ISBN: 978-1-4244-3759-7, pp.576 - 581, Cardiff, Wales, 2009.

[4] Biffl S., Moser T., Winkler D.: Risk Assessment In Multi-Disciplinary (Software+) Engineering Projects, In: International Journal of Software Engineering and Knowledge Engineering (IJSEKE), SI on Software Risk Assessment, 21(2), pp.211-236, 2011.

[5] Biffl S., Halling M.: Investigating the influence of inspector capability factors with four inspection techniques on inspection performance, In: Proceedings of the 8th IEEE Symposium on Software Metrics, pp. 1530-1435, 2002.
[6] Biffl S., Halling M.: Investigating the defect detection effectiveness and cost benefit of nominal inspection teams, In: IEEE Trans. on SW Eng. 29(5), pp. 385-397, 2003.
[7] Conradi R., Mohagheghi P., Arif T., Hegde L.C., Bunde G.A., Pedersen A.: Object-Oriented Reading Techniques for Inspection of UML Models – An Industrial Experiment, In: Proceedings of 17th European Conference on Object-Oriented Programming (ECOOP), LNCS 2743, pp.483-500, 2003.
[8] Fay A., Biffl S., Winkler D., Drath R., Barth M.: A method to evaluate the openness of automation tools for increased interoperability, In: Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON), IEEE, ISBN: 978-1-4799-0223-1, pp.6842-6847, 2013.
[9] Kollanus S., Koskinen J.: Survey of Software Inspection Research: 1991-2005, Working Papers WP-40, University of Jyväskylä, 2007.
[10] Kovalenko O., Winkler D., Kalinowski M., Serral E., Biffl S.: Engineering Process Improvement in Heterogeneous Multi-Disciplinary Environments with the Defect Causal Analysis, In: Proceedings of the 21th EuroSPI, pp.73-85, 2014.
[11] Laitenberger O., DeBaud J-M.: An encompassing life cycle centric survey of software inspection, In: Journal of Systems and Software (JSS), 50(1), pp.5-31, 2000.
[12] Moser T., Mordinyi R., Winkler D., Biffl S.: Engineering Project Management using the Engineering Cockpit: A collaboration platform for project managers and engineers, In: Proceedings of the 9th International Conference on Industrial Informatics (INDIN), 2011.
[13] Moser T., Biffl S., Sunindyo W.D., Winkler D.: Integrating Production Automation Expert Knowledge Across Engineering Domains, In: Int. J. of Distributed Systems and Technologies (IJDST), SI on Emerging Trends and Challenges in Large-Scale Networking and Distributed Systems, 2(3), April-Juni, pp.88-103, 2011.
[14] Porter A., Votta L.: Comparing Detection Methods for Software Requirements Inspection: a Replication Using Professional Subjects, Empirical Software Engineering Journal, 3(4), pp.355-379, 1998.
[15] Robinson B., Francis P., Ekdahl F.: A defect-driven process for software quality improvement, In: Proceedings of ESEM, pp333-335, 2008.
[16] Shull F., Rus I., Basili V.: How perspective-based reading can improve requirements inspection, In: IEEE Computer, 33(7), pp.73-79, 2002.
[17] Thelin T., Andersson C., Runeson P., Dzamashvili-Fogelstrom M.: A replicated experiment of usage-based and checklist-based reading, In: Proceedings of the 10th International Symposium on Software Metrics, pp.246-256, 2004.
[18] Winkler D., Schönbauer M., Biffl S.: Towards Automated Process and Workflow Management: A Feasibility Study on Tool-Supported and Automated Engineering Process Modeling approaches, In: Proc. of Euromicro SEAA, SM Track, pp.102-110, 2014.
[19] Winkler D., Thurnher B., Biffl S.: Early Software Product Improvement with Sequential Inspection Sessions: An Empirical Investigation of Inspector Capability and Learning Effects, Euromicro SEAA, Lübeck, Germany, 2007.
[20] Winkler D.: Improvement of Defect Detection with Software Inspection Variants: A Large-Scale Empirical Study on Reading Techniques and Experience, VDM, 2008.
[21] Winkler D., Biffl S.: Improving Quality Assurance in Automation Systems Development Projects, In: M. Savsar (Ed.): Quality Assurance and Management, INTEC Publishing, ISBN 978-953-51-0378-3, pp.379-398, 2012.
[22] Winkler D., Moser T., Mordinyi R., Sunindyo W., Biffl S.: Engineering Object Change Management Process Observation in Distributed Automation Systems Projects, In: Proceedings of 18th EuroSPI, Industrial Track, 2011.