

Quality Assurance for Data from Low-Tech Participants in Distributed Automation Engineering Environments

Richard Mordinyi, Alexander Pacha, Stefan Biffel

Christian Doppler Laboratory Software Engineering Integration for Flexible Automation Systems
Vienna University of Technology
Favoritenstrasse 9, Vienna, Austria
{firstname.lastname}@tuwien.ac.at

Abstract

Participants in modern automation engineering projects typically work distributed and in parallel. Therefore, there are advanced approaches for integrating the data of specific engineering tools and systems. However, in these projects there are also participants who do not work with the specific engineering tool set but provide important data updates, e.g., customer representatives. A major challenge is to provide systematic and efficient quality assurance for these inputs. In this paper we describe an approach to provide efficient quality assurance when importing data from general purpose tools such as Excel into an integrated engineering data set. We report on experiences with initial prototypes and compare the improved with a traditional data import process to discuss advantages, risks, and further improvements.

1. Introduction

Typical large-scale engineering projects (making e.g., power plants or car manufacturing plants) involve a wide range of engineering disciplines [1], like mechanical, electrical or software engineering. Each of the disciplines bring along their specific engineering tools and engineering systems to manage specific engineering processes which use different technical platforms and heterogeneous data models. However, for building and controlling such projects a cooperation of each discipline is required to form an integrated engineering system. Integration of engineering systems is a challenge as (particularly in the automation industry) typically a broad range of engineering tools from different vendors are used to solve specific problems [2]. Usually, a database is used to integrate data of specific engineering tools and systems.

However, in such projects there are also participants like customer representatives who do not work with those specific engineering tool sets and do not have

access to the integrating database due to security or project management reasons. Nevertheless, such external project partners still provide important data updates and are therefore essential to the success of the engineering project. 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. The extracted spreadsheets are sent to and modified by the external partners and then performed changes are transferred back into the integrated database. 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.

In this paper we describe a work-in-progress approach that provides efficient quality assurance when importing data from general purpose tools like Excel into an integrated engineering database. While there is tool-support for recognition and review of changes, those are neither fully integrated into Excel nor into the process of distribution, reintegration, and quality assurance of engineering data, and thus requires a lot of error-prone and time consuming human effort. A relevant part of the approach is an Excel add-in that contains 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 and hiding negligible information.

2. Related Work

This section summarizes related work on automation systems engineering projects and version management systems.

2.1. Automation Systems Engineering

Automation systems (AS), such as complex industrial automation plants for manufacturing [3] or hydro power

plants [4], depend on distributed software to control systems behavior. In automation systems engineering (ASE) software engineering depends on specification data and plans from a wide range of other engineering aspects in the overall engineering process, e.g., physical plant design, electrical engineering, or process planning. This expert knowledge is embodied in domain-specific standards, terminologies, processes, models, and software tools. Engineering models (e.g., model-based design and testing [5]) help to construct new systems products and to verify and validate the solutions regarding the requirements, specification, and design models. Traditional systems engineering processes follow a waterfall-like engineering process with late testing approaches [6]. Unfortunately, insufficient attention is paid in the field of ASE to capabilities for quality assurance (QA) of software-relevant artifacts and change management across engineering domains [7], possibly due to technical and semantic gaps in the engineering team. Thus, there is considerably higher effort for testing and repair, if defects get identified late in the engineering process.

2.2. Version Management Systems

State-of-the-art revision control systems [8, 9] offer a variety of features to work collaboratively on a single or multiple documents. All revision control systems share the concept of a repository, which is a managed storage for digital data, e.g. source code, documents or data-models. But apart from simply storing the data, a repository contains additional meta-information about the data, like the changes performed on it, or who is allowed to modify the data in combined with access restrictions. Repositories can have different forms, like a centralised repository, where one single server is responsible for everything, or a distributed repository, where every client holds all repository-information.

3. Use Case and Research Issues

Building and controlling large-scale engineering projects requires cooperation not only between the project's groups of different engineering disciplines but also with project external stakeholders who bring valuable input to the project. However, in the traditional way sharing information with external project partners and the reintegration of their modifications into the projects engineering database requires a lot of human and semi-automated process steps with high effort and error-sources.

Figure 1 shows a process model as currently in operation at our industry partner. Whenever, an external project partner asks for project information a specific report is checked out from the integrated database.

The spreadsheet created this way is handed over to the customer. According to our industry partner, exported spreadsheets may contain over 20000 data

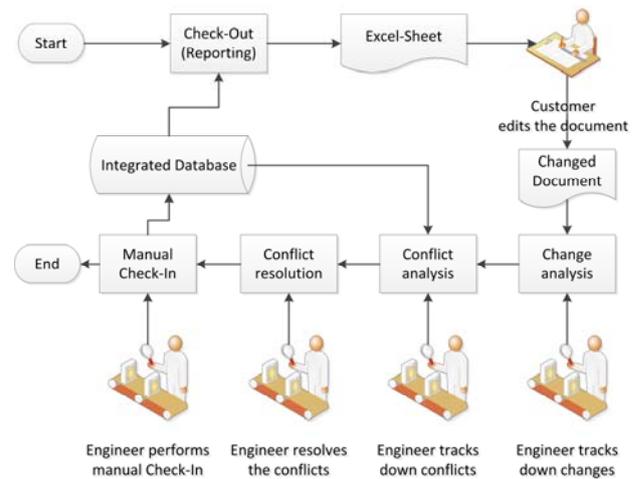


Figure 1. Process diagram describing data exchange with external project partners.

records [10] whereas depending on the state of the project about 20% will be modified [11]. Incoming changes have to be checked; potential conflicts (around 2%) analyzed and resolved.

However, this is a time-consuming task as information about the response for a very specific data record is hard to retrieve. Furthermore, although each of the steps is tool-supported the reviewer(s) of the spreadsheet have to use different tools which are hardly integrated. This results in a chopped up process which has to be conflated manually. Finally, once conflicts have been resolved it is still possible that the integrated database has been altered during the check-in or resolution step resulting in new conflicts which need to be resolved as well.

From this challenge we derive the following key research issues:

RI-1 Change management process: Analyze the current change management process regarding Excel and the integrated engineering database for improvement opportunities. What are the types of defects and risks with respect to re-import of data which require high effort and hinder efficient collaboration with external partners? What are the options for improving quality assurance with respect to the identified error sources?

RI-2 Improved change management process: Investigate to what extent the current process model has to be modified so that it is capable of defect and risk mitigation as well as improving the effectiveness and efficiency of quality assurance. How does the proposed process variant influence the effort for integrating changes into common integrated database?

RI-3 Complexity of engineering artifacts: Compare sources of complexity by evaluating the handling of engineering documents, integrated database and performed modifications on them in the traditional and in the proposed process models. What are the main complexity drivers regarding quality assurance and change management?

RI-4 Tool-support: The efficiency of the proposed process model depends on the extent it can be integrated into already existing engineering tool sets. Therefore, it needs to be evaluated how tool-support for the proposed process variant can be integrated into current tool environments. Investigate its usability by discussing its capabilities and limitations with project administrators, project managers, and external project partners.

4. Proposed Solution Approach

Our proposed approach for improving effectiveness and efficiency of quality assurance with respect to re-importing engineering data modified by external project partners is shown in figure 2.

In contrast to the traditional approach, the proposed solution exports not only Excel spreadsheets, but also enriches it with additional meta and data information, called Shadow Copy. A Shadow Copy is an actual copy of the checked-out set of data and allows engineers to point out changes at any time – even if there is no connection to the integrated database. Meta information contains data about e.g., time of check-out and a list of people detached to the checked-out engineering data who may be contacted in case of question specific to that entry.

This file is then sent to the external project partner who may perform modifications (e.g., adapt cells, switch rows and columns) and review them as necessary. Once finished the spreadsheet is sent back to the project stakeholders who may start re-implementing the received data into the integrated database. The file is first checked for validity to make sure that the version-information is still present and valid. 5. If the meta-information is still present, a comparison between the shadow-copy and the current sheet can be performed to figure out what has changed. Figure 3 shows a screenshot of the spreadsheet. Cells with a yellow background indicate modifications. If the reviewing engineer approves the change, the cell's background color turns green.

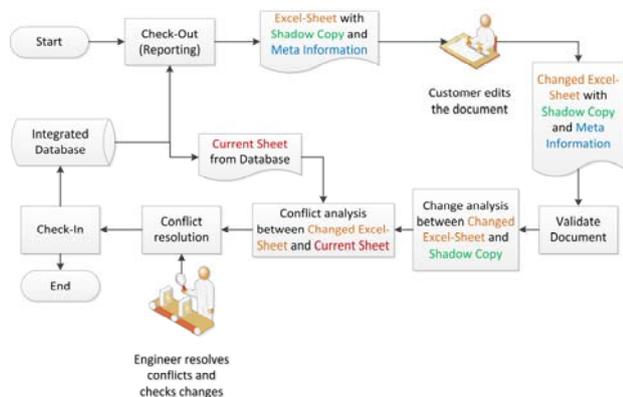


Figure 2. Proposed process diagram with full support for reintegration of engineering data.

The list on the right hand side of the spreadsheet presents all rows which have been modified. Once all changes have been approved the engineer may check-in the spreadsheet and thus re-import retrieved updates. In this the Excel Add-in pulls the current version of the spreadsheet from the integrated database and compares it with the changes approved. As shown in figure 3 conflicting data entries are colored red. Furthermore, the reviewing engineer receives information about the original value of the cell (value at the time of check-out), about the approved changed value, and about the value currently available in the database. Moreover, the spreadsheets displays contact information about the engineer who has modified the data while it has been updated by external partners. Once again, a list on the bottom right hand side of the spreadsheet presents all conflicting rows.

OK	Position	Old	New	Operation
<input checked="" type="checkbox"/>	2;5	CE200	CE201	Modified
<input checked="" type="checkbox"/>	3;5	CE200	CE201	Modified
<input type="checkbox"/>	4;2	Process 1703	Process 1704	Modified
<input type="checkbox"/>	6;4	BAA10	BAA11	Modified
<input type="checkbox"/>	8;4	BAA10	BAA12	Modified
<input type="checkbox"/>	9;8	LNK_SIP	LNK_1703	Modified

Resolved	Position	Old	New	Repository	Operation
<input type="checkbox"/>	7;4	B...	B...	BAA13	Conflict

Figure 3. User interface showing changes and conflicting data records.

In case the engineer starts the check-in process but some of the data to be checked-in is still in conflict with data in the database or even if new conflicts come up, they are pointed out by the spreadsheet immediately.

5. Initial Discussion

In case of the traditional approach engineers collect all engineering data matter to changes and create additional reports. Then they need to check all change reports lines with the data already checked-in into the integrated database. In case of the proposed approach changes are presented in the context of the check-in. Additionally, conflicts are presented in the context of integrated database and the check-in context. This allows grouping by components and provides information about the responsible domain experts for focused discussions.

The goal of the proposed approach is to keep the power and flexibility of Excel without losing the consistency checks of a closed and specific application. The add-in hides the complexity of reintegration of modified data by pointing out changes immediately. Therefore, it is an easy to use to for external project partners as well as for project engineers. The latter is also supported by spotting conflicting data directly and providing suggestions to resolve them without the need

for an extra tool. Due to rapidly faster resolution of conflicts the probability of new conflicts or current during resolution of old conflicts becomes very low. This way the proposed approach provides a continuously tool-supported way of handling modifications and reintegration of them into an existing integrated database.

6. Summary and Further work

Participants in modern automation engineering projects typically work distributed and in parallel using specific engineering tool set. The success of a project also depends on external project partners and the data updates they provide. In this paper we presented a work-in-progress approach which facilitates the efficient reintegration of engineering data updates coming from external project partners. The approach minimizes human effort in identifying data changes and data conflicts by providing consistent tool-support in Excel spreadsheets. The first results are promising as we have gained good experiences with the current prototype in a limited context in the lab. Further work will focus on the detailed investigation and evaluation of the described research issues in the context of a running engineering project of our industry partner.

Acknowledgements

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

References

- [1] Biffl, S. 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. p. 75 - 92.
- [2] Rangan, R.M., S.M. Rohde, R. Peak, B. Chadha, and P. Bliznakov, *Streamlining Product Lifecycle Processes: A Survey of Product Lifecycle Management Implementations, Directions, and Challenges*. *Journal of Computing and Information Science in Engineering*, 2005. **5**: p. 227-237.
- [3] Biffl, S., W.D. Sunindyo, and T. Moser, *Bridging Semantic Gaps Between Stakeholders in the Production Automation Domain with Ontology Areas*, in *21st International Conference on Software Engineering and Knowledge Engineering*. 2009, KSI. p. 233-239.
- [4] Schindele, A., *logi.CAD Anwendungen in Wasserkraftanlagen*. Presentation at logi.cals Powerdays, Monheim, Germany, 2009.
- [5] Baker, P., R.D. Zhen, J. Gabrowksi, and H. Oystein, *Model-Driven Testing: Using the UML Testing Profile*. 2007: Springer.
- [6] *GAMP 5 - Good Automated Manufacturing Practice*. International Society for Pharmaceutical Engineering (ISPE), 2008.
- [7] Schäfer, W. and H. Wehrheim, *The Challenges of Building Advanced Mechatronic Systems*, in *2007 Future of Software Engineering - International Conference on Software Engineering*. 2007, IEEE Computer Society: Washington, DC. p. 72-84.
- [8] O'Sullivan, B., *Making Sense of Revision-control Systems*. *Queue*, 2009. **7**(7): p. 30-40.
- [9] Spinellis, D., *Version control systems*. *Software*, IEEE, 2005. **22**(5): p. 108-109.
- [10] Moser, T., R. Mordinyi, D. Winkler, and S. Biffl, *Engineering Project Management Using The Engineering Cockpit*, in *IEEE 9th International Conference on Industrial Informatics (INDIN'2011)*. 2011: Lisbon, Portugal. p. accepted for publication.
- [11] Winkler, D., T. Moser, R. Mordinyi, W.D. Sunindyo, and S. Biffl, *Engineering Object Change Management Process Observation in Distributed Automation Systems Projects*, in *18th European System & Software Process Improvement and Innovation (EuroSPI 2011)*. 2011: Denmark. p. accepted for publication.