Utilizing Data Analytics to Support Process Implementation in Knowledge-intensive Domains

Gregor Grambow
Computer Science Dept.
Aalen University
Aalen, Germany
e-mail: gregor.grambow@hs-aalen.de

Abstract— **In recent times, knowledge-intensive activities and processes have become more and more important in various areas like new product development or scientific projects. Such processes are hard to plan and control because of their high complexity, dynamicity, and human involvement. This imposes numerous threats to successful and timely project execution and completion. In this paper, we propose an approach to support such processes and projects holistically. The basic idea is to utilize various kinds of data analytics on different data sets, reports, and events occurring in a project. This data can be used to fill the gap between the abstract process planning and its dynamic operational enactment. That way, processes can be technically implemented and supported in complicated knowledge-intensive domains and also adapted to changing situations.**

Keywords-data analytics; knowledge-intensive projects; process implementation.

I. INTRODUCTION

In the last decades, the number and importance of knowledge-intensive activities has rapidly increased in projects in various domains [1][2]. Recent undertakings involving the inference of knowledge utilizing data science and machine learning approaches also require the involvement of humans interpreting and utilizing the data form such tools. Generally, knowledge-intensive activities imply a certain degree of uncertainty and complexity and rely on various sets of data, information, and knowledge. Furthermore, they mostly depend on tacit knowledge of the humans processing them. Hence, such activities constitute a huge challenge for projects in knowledge-intensive domains, as they are mostly difficult to plan, track and control.

Typical examples for the applications of such activities are business processes in large companies [1], scientific projects [3], and projects developing new products [4]. In each of these cases, responsibles struggle and often fail to implement repeatable processes to reach their specific goals.

In recent times, there has been much research on data storage and processing technologies, machine learning techniques and knowledge management. The latter of these has focused on supporting whole projects by storing and disseminating project knowledge. However, projects still lack a holistic view on their contained knowledge, information and data sets. There exist progressive approaches for storing data and drawing conclusions from it with statistical methods or neural networks. There also exist tools and methods for organizing the processes and activities of the projects. Nevertheless, in most cases, these approaches stay unconnected. Processes are planned, people execute complex tasks with various tools, and sometimes record their knowledge about procedures. However, the links between these building blocks stay obscured far too often.

In this paper, we propose a framework that builds upon existing technologies to execute data analyses and exploit the information from various data sets, tools, and activities of a project to bring different project areas closer together. Thus, the creation, implementation, and enactment of complex processes for projects in knowledge-intensive domains can be supported.

The remainder of this paper is organized as follows: Section II provides background information including an illustrating scenario. Section III distils this information into a concise problem statement. Section IV presents an abstract framework as solution while Section V provides concrete information on the modules of this framework. This is followed by an evaluation in Section VI, related work in Section VII, and the conclusion.

II. BACKGROUND

In the introduction, we use the three terms data, information and knowledge. All three play an important role in knowledge-intensive projects and have been the focus of research. Recent topics include research on knowledge management and current data science approaches. Utilizing definitions from literature [5], we now delineate these terms in a simplified fashion:

- Data: Unrefined factual information.
- Information: Usable information created by organizing, processing, or analyzing data.
- Knowledge: Information of higher order derived by humans from information.

This taxonomy implies that information can be inferred from data manually or in a (semi-)automated fashion while knowledge can only be created by involving the human mind. Given this, knowledge management and data science are two fields that are complementary. Data science can create complex information out of raw data while knowledge management helps the humans to better organize and utilize the knowledge inferred from that information.

Processes in knowledge-intensive domains have special properties compared to others, like simple production processes [6]. They are mostly complex, hard to automate, repeatable, can be more or less structured and predictable

 ISBN: 978-1-61208-681-1

and require lots of creativity. As they are often repeatable, they can profit from process technology enabling automated and repeatable enactment [7].

In the introduction, we mentioned three examples for knowledge-intensive processes: scientific projects, business processes in large companies and new product development. We will now go into detail about the properties of these.

In scientific projects, researchers typically carry out experiments generating data from which they draw knowledge. The amount of processed data in such projects is rapidly growing. To aid these efforts, numerous technologies have been proposed, on the one hand for storage and distributed access to large data sets. On the other hand, many frameworks exist supporting the analysis of such data with approaches like statistical analyses or neuronal networks [8]. There also exist approaches for scientific workflows enabling the structuring of consecutive activities related to processing the data sets [9]. However, the focus of all these approaches is primarily the processing of the scientific data. A holistic view on the entire projects connecting these core activities with all other aspects of the projects is not prevalent. In addition, the direct connection from data science to knowledge management remains challenging.

Business processes in large companies are another example of knowledge-intensive processes. Such processes are often planned on an abstract level and the implementation on the operational level remains difficult due to numerous special properties of the context of the respective situations. Consider a scenario where companies work together in complex supply chains to co-create complex products like in the automotive industry. Such companies have to share different kinds of information. However, this process is rather complicated as the supply chains are often huge with hundreds of participants. A data request from the company at the end of the chain can result in thousands of recursive requests through the chain [10]. For each request, it must be separately determined, which are the right data sets that are needed and can be shared.

A third example are projects developing new products. As example, we focus on software projects because software projects are essentially knowledge-intensive projects [4]. For these, various tools exist from development environments to tools analyzing the state of the source code. In addition to this, usually a specified process is also in place. However, the operational execution relies heavily on individuals that have to analyze various reports and data sources manually to determine the correct course of action in order to create high quality software. This implies frequent process deviations or even the complete separation of the abstract planned process from its operational execution. Furthermore, due to the large amount of available data sets (e.g., specifications, bug reports, static analysis reports) things may be forgotten and incorrect decisions made.

Figure 1 illustrates different problems occurring when trying to implement a software development process on the operational level. In particular, it shows an excerpt of an agile software development process (the Open UP). The process comprises the four phases Inception, Elaboration, Construction, and Transition. Each of these, in turn, comprises an arbitrary number of iterations. Each iteration contains different concrete workflows to support activities like requirements management or software development. As an example, we show the 'Develop Solution Increment' workflow that covers operational software development. It contains concrete activities like 'Implement Solution' where the developer shall technically implement the solution (i.e., a specific feature of a software), which was designed before. However, such activities are still rather abstract and have no connection to tasks the human performs to complete the activity. These tasks are performed with concrete tools, artifacts, and other humans depicted in the blue box of Figure 1. The figure indicates various issues: (1) Such tasks performed with different tools like Integrated Development Environments (IDEs) and static analysis tools are fine-grained and dynamic. Therefore, the workflow cannot prescribe the exact tasks to be performed [11]. Furthermore, the mapping of the numerous real world events to the workflow activities is challenging. (2) In various situations, the developer must derive decisions based on data contained in reports from different tools. One example are specific changes to improve the source code to be applied on account of static analysis reports. Goal conflicts (e.g., high performance vs. good maintainability) may arise resulting in wrong decisions. (3) In various cases, different artifacts (e.g., source code and code specifications) may relate to each other and can be processed simultaneously by different persons, which may result in inconsistencies [12].

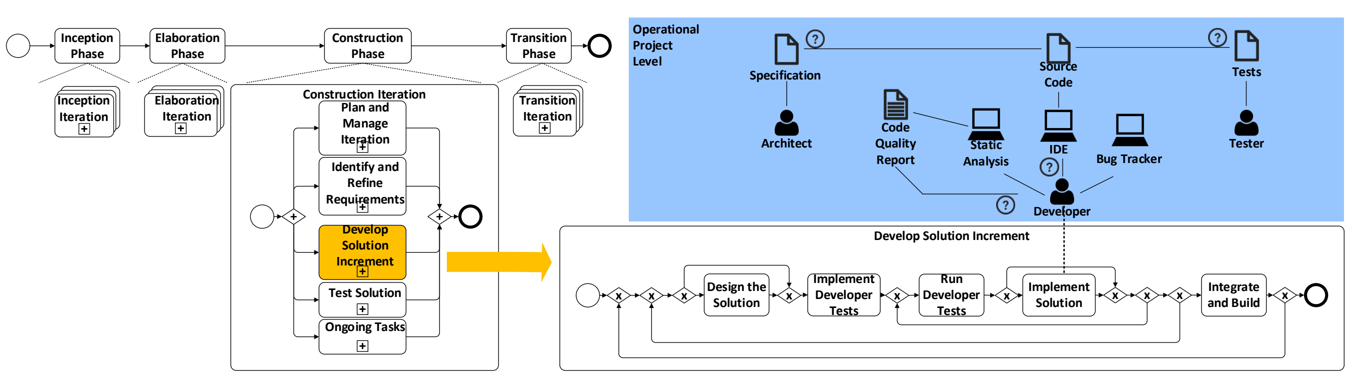

Figure 1. Scenario.

 ISBN: 978-1-61208-681-1

(4) Unexpected situations may lead to exceptions and unanticipated process deviations. (5) The whole process relies on knowledge. Much of this knowledge is tacit and is not captured to be reused by other persons [13]. This often leads to problems.

III. PROBLEM STATEMENT

In Section II, we have defined different kinds of relevant information and shown examples from different domains in which a lacking combination of such information leads to problems with operational process implementation.

In scientific projects, data analysis tools aid humans in discovering information in data. However, the projects mostly neither have support for creating, retaining, and managing knowledge derived from that information, nor do they have process support beyond the data analysis tasks [13][14]. Complex business processes in large companies often suffer from lacking process support because of the high number of specific contextual properties of the respective situations. In new product development, problems often arise due to the inability to establish and control a repeatable process on the operational level. This is caused by the high number of dynamic events, decisions, deviations, and goal conflicts occurring on the operational level.

In summary, it can be stated that process implementation in knowledge-intensive projects is problematic due to the high complexity of the activities and relating data. Processes can be abstractly specified but not exactly prescribed on the operational level. Thus, it remains difficult to track and control the course of such projects which often leads to exceeded budgets and schedules and even failed projects.

IV. FRAMEWORK

In this paper, we tackle these challenges by proposing an approach uniting different kinds of data analytics and their connection to other project areas like knowledge management and process management. That way we achieve a higher degree of automation supporting humans in their knowledge-intensive tasks and facilities to achieve holistic and operational implementation of the projects process.

Because of the high number of different data sets and types and their impact on activities, we think it is not possible to specify a concrete framework suitable for all possible use cases of knowledge-intensive projects of various domains. We rather propose an extensible abstract framework and suggest different modules and their connections based on the different identified data and information types in such projects. The idea of this abstract framework builds on our previous research where we created and implemented concrete frameworks for specific use cases. Hence, we use our experience to extract general properties from these frameworks to achieve a broader applicability.

The basic idea of such a framework is a set of specific modules capable of analyzing different data sets and utilizing this for supporting knowledge-intensive projects in various ways. Each of these modules acts as a wrapper for a specific technology. The framework, in turn, provides the following basic features and infrastructure to foster the collaboration of the modules.

A simple communication mechanism. The framework infrastructure allows each module to communicate with the others to be able to receive their results and provide its results to the others.

Tailoring. The organization in independent modules facilitates the dynamic extension of the framework by adding or removing modules. That way the framework can be tailored to various use cases avoiding technical overhead.

Support for various human activities. The framework shall support humans with as much automation as possible. Activities that need no human intervention shall be executed in the background providing the results in an appropriate way to the humans. In contrast to this, activities that require human involvement shall be supported by the framework. All necessary information shall be presented to the humans helping them to not forget important details of their tasks.

Holistic view on the project. Various technologies for different areas of a project are seamlessly integrated. That way, these areas, like process management, data analysis, or knowledge management can profit from each other.

Process implementation. The framework shall be capable of implementing the process spanning from the abstract planning to the operational execution.

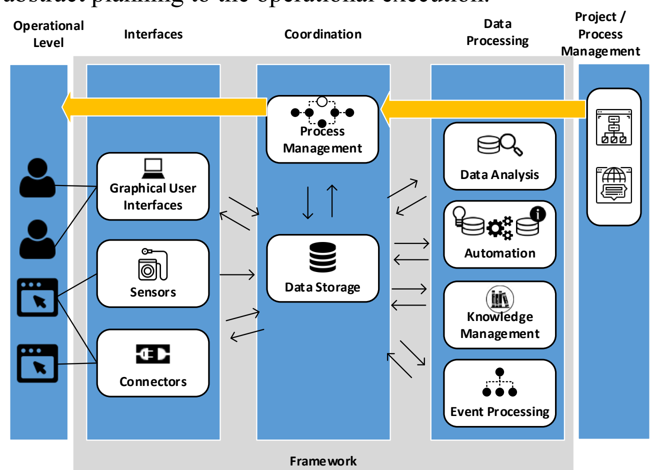

Figure 2. Abstract Framework.

Figure 2 illustrates the framework. We divide the latter into three categories of modules: Interfaces, Coordination, and Data Processing. The coordination category contains the modules responsible for the coordination of data and activities in the framework: The data storage module is the basis for the communication of the other modules by storing and distributing the messages between the other components. The process management module is in charge of implementing and enacting the process. Thus, it contains the technical representation of the processes specified at the project / process management level, which is outside the framework. Utilizing the other modules, these processes can be enacted directly on the operational level where concrete persons interact with concrete tools. This improves repeatability and traceability of the enacted process.

The interface category is comprised of three modules: Graphical user interfaces enable users to communicate with the framework directly, e.g., for controlling the process flow

 ISBN: 978-1-61208-681-1

or storing and utilizing knowledge contained in the framework. The sensor module provides an infrastructure for receiving events from sensors that can be integrated into external software tools or from sensors from production machines. That way, the framework has access to real-time event data from its environment. The connector module provides the technical interface to communicate with APIs of external tools to exchange data with the environment.

The data processing category provides the following modules: The event processing module aggregates event information. This can be used, for example, for determining actions conducted in the real world. Therefore, sensor data from the sensor module can be utilized. By aggregating and combining atomic events, new events of higher semantic value can be generated. The data analysis module integrates facilities for statistical data analytics and machine learning. This can be utilized to infer information from raw data, e.g., coming from production machines or samples in scientific projects. The knowledge management component aids humans in managing knowledge derived from it. Both technologies can interact to support scientific workflows. E.g., incoming data can be analyzed and classified and the framework can propose an activity to a human for reviewing the data and record knowledge in a knowledge base.

Finally, the automation component enhances the automation capabilities of the framework. Therefore, various technologies are possible. As a starting point, we propose the following: rules engines for simple specification and execution of rules applying for the data or the project as a whole. One example use case is the automated processing of reports from external tools. Multiple reports can be processed creating a unified report by a rules-based transformation that, in turn, can be processed by other modules. A second important technology for automation are multi-agent systems. They enhance the framework by adding automated support for situations with goal conflicts. Consider situations where deviations from the plan occur and the framework shall determine countermeasures. Software refactoring is one possible use case: When the framework processes reports of static analysis tools indicating quality problems in the source code, software quality measures can help. However, mostly there are too many problems to tackle all and the most suitable must be selected. In such situations, agents perusing different quality goals like maintainability or reliability can autonomically decide on software quality measures that are afterwards integrated into the process in cooperation with the other modules [11].

V. MODULES

This section provides details on the different modules, their capabilities and the utilized technologies.

Data Storage. As depicted in Section IV, the first use case for this module is being the data store for the module communication. Messages are stored here and the modules can register for different topics and are automatically notified if new messages are available for the respective topic. This also provides the basis for the loose-coupling architecture. However, this module is not limited to one database technology but enables the integration of various technologies to fit different use cases. One is the creation of a project ontology using semantic web technology to store and process high-level project and domain knowledge that can be used to support the project actors.

Process Management. This module provides PAIS (Process-Aware Information System) functionality: Processes are not only modelled externally at the project management level as an idea of how the project shall be executed but can be technically implemented. Thus, the enactment of concrete process instances enables the correct sequencing of technical as well as human activities. Humans automatically receive activities at the right time and receive support in executing these. To enable the framework to react on dynamic changes we apply adaptive PAIS technology [15]. That way the framework can automatically adapt running process instances. Consider an example from software development projects: Software quality measures can be inserted into the process automatically when the framework detects problems in the source code by analyzing reports from static analysis tools [11]. This actively supports software developers in achieving better quality source code.

Sensors. This module comprises facilities for receiving events from the frameworks environment. These events can be provided by hardware sensors that are part of production machines. This can also be established on the software side by integrating sensors in the applications used by knowledge workers. That way, information regarding the processed artifacts can be gathered. Examples regarding our scenario from Section II include bug trackers and development tools so the framework has information about bugs in the software and the current tasks developers process.

Graphical User Interfaces. GUIs enable humans to interact with the framework directly. Firstly, this applies to the enactment of processes with the framework. The latter can provide activity information to humans guiding them through the process. In addition, humans can control the process via GUIs indicating activity completion and providing the framework with information on their concrete work. Another use case is storing knowledge in a knowledge store being part of the framework. To enable this, the GUI of a semantic wiki integrated into the framework as knowledge store can be exposed to let humans store the knowledge and annotate it with machine-readable semantics. That way, the framework can provide this knowledge to other humans in an automated fashion. However, GUIs are also used for configuring the framework to avoid hard-coding its behavior matching the respective use case. One example is a GUI letting humans configure the rules executed in the integrated rules engine. Thus, e.g., it can be configured which parts of external reports shall be used for transformation to a unified report the framework will process.

Connectors. This module is applied to enable technical communication with external tools. Depending on the use case, interfaces can be implemented to call APIs of other tools or to be called by these. Consider an example relating to the projects' process: The process is externally modeled utilizing a process modeling tool. This process can be transformed (manually or automatically) to a specification our framework uses for process enactment. In the process

 ISBN: 978-1-61208-681-1

enactment phase, the external tool can be automatically updated displaying the current state of execution.

Automation. For this module we proposed two technologies as a starting point: rules engines can be utilized for simple automation tasks. One use case is, as mentioned, automatic transformation of reports from multiple external tools into one unified report. Multi-agent systems are applicable in situations where goals conflicts apply. Consider the example regarding the quality of newly created software: In software projects, often multiple static analysis tools are executed providing metrics regarding the source code quality. Usually, there is not enough time to get rid of all issues discovered. It is often challenging for software engineers to determine the most important software quality measures to be applied. Such projects mostly have defined quality goals as maintainability or reliability of the source code. Quality goals can be conflicting as, e.g., performance and maintainability and different measures support different quality goals. For such situation, agents can be applied: Each goal gets assigned an agent with a different strategy and power. When a quality measure can be applied the agents utilize a competitive procedure for determining the most important quality measure to be applied.

Data Analysis. This module enables the integration of frameworks or libraries for statistical analysis and machine learning approaches like Scikit-learn [8]. The advantage of the integration in the framework infrastructure is option to execute such tools as part of a holistic process. Data that has been acquired by other modules can be processed and the results can also be stored in the frameworks data storage. Furthermore, other modules can be notified so humans can be involved. For example a process can be automatically executed where data is analyzed and the results are presented to humans that, in turn, can derive knowledge from them and directly manage this knowledge with the knowledge management component.

VI. EVALUATION

We now provide two concrete scenarios in which we have created and successfully applied concrete frameworks that implement our idea of this abstract framework. The first one comes from the software engineering domain. For this domain, we have implemented a comprehensive framework including all of the mentioned modules [11][12][14].

This includes the implementation of a key-value store for framework communication on top of an XML database. The latter was also used to store numerous reports from internal and external tools natively. We further applied the AristaFlow BPM suite for process implementation and adaptation and recorded all high level project information in an OWL ontology. Data acquisition was realized by connectors to tools like bug trackers or project management tools and a sensor framework enabling the integration of sensors in tools like Eclipse or Visual Studio. Thus, we recorded events like saving files, which were aggregated via complex event processing to gather information about what humans were working on. Combining this with an integrated rules engine and a multi-agent system, we realized various use cases. One of them was the automatic provision of software quality measures. The framework automatically received reports from static code analysis tools that were transformed into one unified report, which was analyzed by autonomous agents pursuing different quality goals. Via the goal question metric technique they related the problems and quality measures to their goals and chose quality measures to be automatically integrated into developers' workflows. Another use case was activity coordination: with the project ontology we determined relations of different artifacts and could automatically issue follow-up activities for example to adapt a software specification if the interface of a components' source code was changed and vice versa. The integration of a semantic wiki enabled the following: Knowledge was recorded and annotated by humans and thus, the framework could automatically inject this knowledge into the process to support other humans in similar activities. In this project, we applied the framework in two SMEs and successfully evaluated its suitability.

The second scenario involves a business use case in which different companies in a supply chain had to exchange sustainability information regarding their production [10]. The producer of the end product has to comply with many laws and regulations and must collect information from the whole supply chain resulting in thousands of recursive requests. On the operational level, this process is very complex as it is difficult to determine which information is important for sustainability, which one must be externally evaluated to comply, and which information should not be shared as it reveals internals about the production process. To implement such data exchange processes automatically, we applied a more tailored-down version of our framework [16]: The focus were contextual properties that have an influence on the data collection processes. These were modeled in the framework and could be obtained from the frameworks' environment by GUIs and connectors. By analyzing these properties and using the results to adapt processes, we were able to automatically create customized data exchange processes suiting different situations. Due to the size of the supply chain, we combined a content repository for the different documents being exchanged with an in-memory key-value store. In this project, the framework was evaluated by a consortium of 15 companies and was later transferred to one of them to build a commercial tool from it.

These slightly different scenarios demonstrate the advantages of our approach: Its modules can be implemented matching the use case. The framework facilitates the communication between the modules and enables not only data analyses but also automated actions resulting from these supporting process and knowledge management.

VII. RELATED WORK

To the best of our knowledge, there exists no directly comparable approach enabling holistic integration of various data analysis capabilities to support and operationally implement processes in knowledge-intensive domains. However, in different domains, there exist approaches to support projects and processes. One example are scientific workflow management systems [3][9]. Such systems support projects in the processing of large amounts of data. Their focus is the organization and parallelization of data-intensive tasks. Hence, they support the different steps taken to analyze data sets but are not able to support whole projects.

In the software engineering (SE) domain, there have also been numerous efforts to support projects and their processes. Early approaches include the Process-centered Software Engineering Environments (PCSEEs) [17][18]. These environments supported different SE activities and made process enactment possible. However, their handling was complex and configurability was cumbersome what made them obsolete. More recent approaches also exist but these frameworks focused on a specific areas of the projects. Examples are artifact-based support [19] and model-driven approaches [20]. Hence, these frameworks could not provide holistic support for entire projects.

The business domain also features complex knowledge-intensive processes. However, this domain is dominated by tools focusing on the processed data like ERP systems or specialized tools. One concrete example regarding the aforementioned sustainability data use case is BOMcheck [21], a tool that helps companies handling sustainability data. In particular, this tool contains current sustainability information on various materials but is not capable of supporting the process of data handling and exchange.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we presented a broadly applicable approach to support process implementation in knowledge-intensive domains. Based on our experience from prior research projects we suggested an extensible set of modules whose collaboration enables holistic support for projects. Furthermore, we proposed technologies, frameworks and paradigms to realize these modules with specific properties.

We have shown problems occurring in projects in different knowledge-intensive domains and provided an illustrative example from the software engineering domain. Such problems are mostly related to operational dynamics, complex data sets, and tacit knowledge. Our framework enables automatic processing of various data sets relating to the activities in such projects to not only support these activities but also their combination to a knowledge-intensive process. Thus, humans can be supported in transforming data to information and information to knowledge.

Finally, as evaluation, we have shown two concrete cases were we have successfully implemented such a framework in different domains. As future work, we plan to extend the set of modules of our framework and to extend the technology options to realize these modules. We also want to specify concrete interfaces of the modules to enable standardized application and easy integration of new technologies. Finally, we plan to specify types of use cases and their mapping to concrete manifestations of our framework.

ACKNOWLEDGMENT

This work is based on prior projects at Aalen and Ulm Universities in cooperation with Roy Oberhauser and Manfred Reichert.

REFERENCES

[1] M. P. Sallos, E. Yoruk, and A. García-Pérez, “A business process improvement framework for knowledge-intensive entrepreneurial ventures,” The J. of Technology Transfer 42(2), pp. 354–373, 2017.

[2] O. Marjanovic and R. Freeze, “Knowledge intensive business processes: theoretical foundations and research challenges,” HICSS 2011, pp. 1-10, 2011.

[3] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, “A survey of data-intensive scientific workflow management,” J. of Grid Computing 13(4), pp. 457-493, 2015.

[4] P. Kess and H. Haapasalo, “Knowledge creation through a project review process in software production,” Int'l J. of Production Economics, 80(1), pp. 49-55, 2002.

[5] A. Liew, “Understanding data, information, knowledge and their inter-relationships,” J. of Knowl. Manag. Practice 8(2),pp. 1-16, 2007.

[6] O. Isik, W. Mertens, and L. Van den Bergh, “Practices of knowledge intensive process management: Quantitative insights,” BPM Journal, 19(3), pp. 515-534, 2013.

[7] F. Leymann and D. Roller, Production workflow: concepts and techniques. Prentice Hall, 2000.

[8] G. Varoquaux et al.: Scikit-learn, “Machine learning without learning the machinery,” GetMobile: Mobile Computing and Communications, 19(1), pp. 29-33, 2015.

[9] B. Ludäscher et al., “Scientific workflow management and the Kepler system,” Concurrency and Computation: Practice and Experience, 18(10), pp. 1039-1065, 2006.

[10] G. Grambow, N. Mundbrod, J. Kolb, and M. Reichert, “Towards Collecting Sustainability Data in Supply Chains with Flexible Data Collection Processes,” SIMPDA 2013, Revised Selected Papers, LNBIP 203, pp. 25-47, 2015.

[11] G. Grambow, R. Oberhauser, and M. Reichert, “Contextual injection of quality measures into software engineering processes,” Int'l J. on Advances in Software, 4(1 & 2), pp. 76-99, 2011.

[12] G. Grambow, R. Oberhauser, and M. Reichert, “Enabling automatic process-aware collaboration support in software engineering projects,” Selected Papers of ICSOFT'11, CCIS 303, pp. 73-89, 2012.

[13] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel, “Semantic wikis,” IEEE Software, 25(4), pp. 8-11, 2008.

[14] G. Grambow, R. Oberhauser, and M. Reichert, “Knowledge provisioning: a context-sensitive processoriented approach applied to software engineering environments,” Proc 7th Int'l Conf. on Software and Data Technologies, pp. 506-515, 2012.

[15] P. Dadam and M. Reichert: The ADEPT Project, “A Decade of Research and Development for Robust and Flexible Process Support - Challenges and Achievements,” Computer Science - Research and Development, 23(2), pp. 81-97, 2009.

[16] N. Mundbrod, G.Grambow, J. Kolb, and M. Reichert, “Context-Aware Process Injection: Enhancing Process Flexibility by Late Extension of Process Instances,” Proc. CoopIS15, pp. 127-145, 2015.

[17] S. Bandinelli, A. Fuggetta, C. Ghezzi, L. Lavazza, “SPADE: an environment for software process analysis, design, and enactment,” Software Process Modelling and Technology. Research Studies Press Ltd., pp. 223-247, 1994.

[18] R. Conradi, C. Liu, and M. Hagaseth, “Planning support for cooperating transactions in EPOS,” Information Systems, 20(4), pp. 317-336, 1995.

[19] A. de Lucia, F. Fasano, R. Oliveto, and G. Tortora, “Fine-grained management of software artefacts: the ADAMS system,” Software: Practice and Experience, 40(11), pp. 1007-1034, 2010.

[20] F. A. Aleixo, M. A. Freire, W. C. dos Santos, U. Kulesza, “Automating the variability management, customization and deployment of software processes: A model-driven approach,” Enterprise Information Systems, pp. 372-387, 2011.

[21] BOMcheck: https://www.bomcheck.net. [retrieved 09, 2018]