# Achieving Higher-level Support for Knowledge-intensive Processes in Various Domains by Applying Data Analytics

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

*Abstract* — **In many domains like new product development, scientific projects, or complex business cases, knowledge-intensive activities and processes have gained high importance. Such projects are often problematic and may suffer from various threats to successful and timely project completion. This is often caused by the involved knowledge-intensive processes because of their high dynamicity, complexity, and complex human involvement. In this paper, we describe an abstract framework capable of managing and supporting such projects holistically. This is achieved by applying various kinds of data analytics on the different data sets being part of the projects. Thus, processes can be implemented and supported technically utilizing the results and combinations of the data analytics. We furthermore illustrate the applicability of the abstract framework by describing two concrete implementations of this framework in two different domains.**

***Keywords-data analytics; knowledge-intensive processes; process implementation; knowledge management***

## I. Introduction

This paper is an extension of the article “Utilizing Data Analytics to Support Process Implementation in Knowledge-intensive Domains” [1]. It adds a comprehensive evaluation of the approach describing two concrete technical applications of the envisioned framework in detail as well as an extended discussion of related work and extended scenarios, figures and explanations. In the last decades, the number and importance of knowledge-intensive activities has rapidly increased in projects in various domains [2][3]. 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. According to literature [4][5], knowledge-intensive processes are characterized as follows:

- They are a composition of prospective activities whose execution contributes to achieving a certain goal.
- They rely on knowledge workers performing interconnected knowledge-intensive activities.
- They are knowledge-, information-, and data-centric.
- They require substantial flexibility, at design- and run-time.

Typical examples for the applications of such activities and processes are business processes in large companies [2], scientific projects [6], and projects developing new products [7]. 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 [8], 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 [9]. They are mostly complex, hard to automate, repeatable, can be more or less structured and predictable and require lots of creativity. As they are often repeatable, they can profit from process technology enabling automated and repeatable enactment [10].

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 [11]. There also exist approaches for scientific workflows enabling the structuring of consecutive activities related to processing the data sets [12]. 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 [13]. 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 [7]. 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.

We will now illustrate different problems occurring when trying to implement a software development process on the operational level. Therefore, we will utilize an agile software development process: the OpenUP. The process comprises the four phases Inception, Elaboration, Construction, and Transition as illustrated in Figure 1.

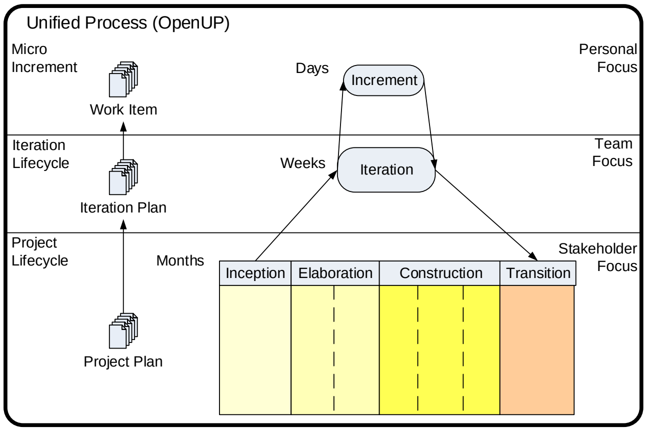

Figure 1. Software Development Process.

These phases cover the entire project lifecycle and are executed with a stakeholder focus. Each of the phases, in turn, may comprise an arbitrary number of iterations. In the latter, the focus lies on the team managing the scope of the iteration with an iteration plan. Each iteration contains different concrete workflows to support activities like requirements management or software development. Each participating person processes the concrete activities of these workflows working on one or more work items. The project lifecycle is managed in the granularity of months, the iterations are more fine grained. Finally, the processing of the work items is done on a daily basis. However, besides various concrete workflows and activities there are also various artifacts, tools, roles, and persons involved. We will now provide details on the OpenUP, its implementation, and issues regarding to it on the operational level as depicted in Figure 2. The figure shows the workflows of the iterations of the four phases. Each of the activities within these represents a sub-workflow containing more fine-grained activities. Every iteration has a sub-workflow for managing the iteration containing activities for planning, managing and assessing the iteration.

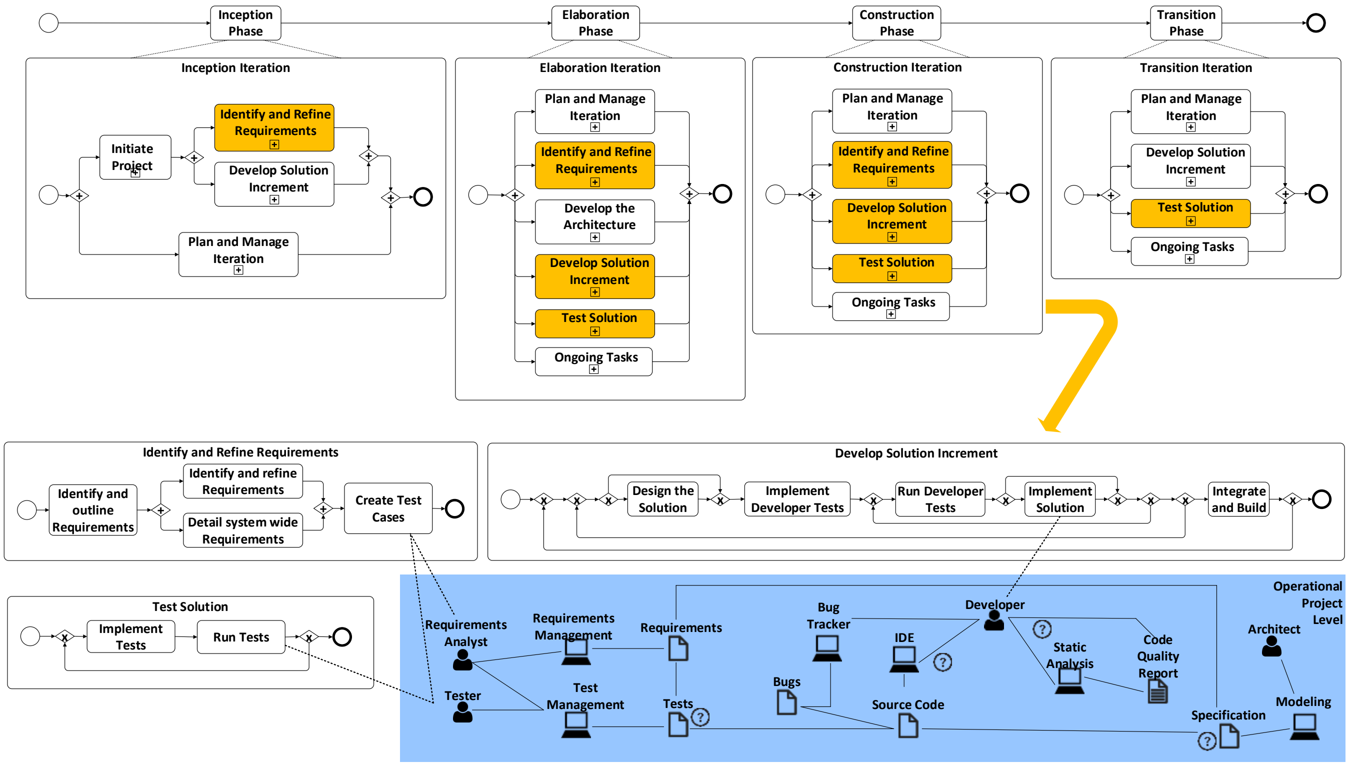

Figure 2. Scenario.

The iterations for all but the first phase also have a sub-workflow called ‘Ongoing Tasks’ for managing changes, e.g., in case the scope or the requirements change. The inception phase primarily deals with setting up the project and the requirements but also allows for creating the first increment of the envisioned solution. The elaboration phases’ iterations add activities for creating the architecture of the software and already the first testing while refining the requirements and continuing to create the solution. The construction phase, in turn, is the main development phase. In the transition phase, the development and testing is finalized to transfer the software to the client. In this phase no more requirement changes shall take place.

As examples, we also show three concrete workflows: ‘Identify and refine Requirements’ deals with the initial creation and refinement of the requirements. In addition, system wide technical requirements are detailed and the relating test cases must be created. ‘Develop Solution Increment’ 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. ‘Test Solution’ contains a loop of creating and running tests for the created software. However, such activities are still rather abstract and have no connection to tasks the human performs to complete the activities. These tasks are performed with concrete tools, artifacts, and other humans depicted in the blue box of Figure 2. The figure indicates various issues: (1) Tasks performed with different tools like IDEs and static analysis tools are fine-grained and dynamic. Therefore, the workflow cannot prescribe the exact tasks to be performed [14]. 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 technical specifications) that relate to each other may be processed simultaneously by different persons, which may result in inconsistencies [15]. (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 [16]. 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 [16][17]. 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.

In particular, the following points need to be addressed:

- Seamless integration of data analysis approaches into the projects. Data producers, data storage and data consumers should be integrated globally in projects.
- Integration of (semi-)automated data analytics with knowledge management.
- Integration of data analytics with automated process support to automatically adapt the process to changing situations.

## 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.

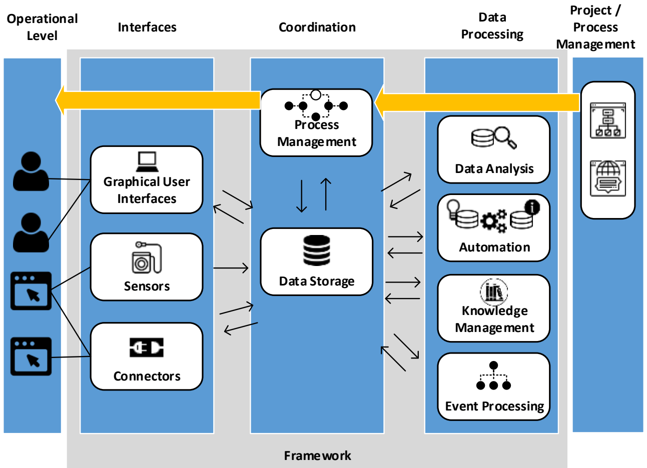

Figure 3. Abstract Framework.

**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.

In the following, the structure of the framework and the interplay of its components are described. A more concrete description of each component follows in Section V. The framework is illustrated by Figure 3. 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 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 modules relating to data processing and analytics, which enables the framework to automatically issue various actions and influence the process to fit to changing situations: 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 [14].

## 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 [18]. 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 [14]. 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 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 semantic reasoning, statistical analysis, or machine learning frameworks like Scikit-learn [11]. 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. That way, data analysis approaches can be seamlessly integrated at several points in the process to achieve a higher level of support and automation.

## 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 [14][15][17] as illustrated in Figure 4.

The framework is based on a loose-coupling architecture where different modules managed by an OSGI [19] infrastructure that communicate via events. Such events are stored and distributed by the data storage module. The latter is realized by a key-value store based on the XML database eXist [20]. The events are organized in several collections for which the modules can register to be automatically notified in case of new events regarding a certain topic. Communication with the frameworks environment is realized on the one hand by web-based GUIs and connectors to tools like bug trackers. On the other hand, the event extraction module applies the framework Hackystat [21] to be able to integrate sensors on various tools like IDEs or source control management tools. These sensors generate events in several situations like opening files or perspective switches in the IDEs. The events are sent to the data storage component, which, in turn, provides them to the event processing component. The latter applies complex event processing (CEP) utilizing the tool Esper [22]. That way, events with higher semantic value can be generated out of multiple low level events.

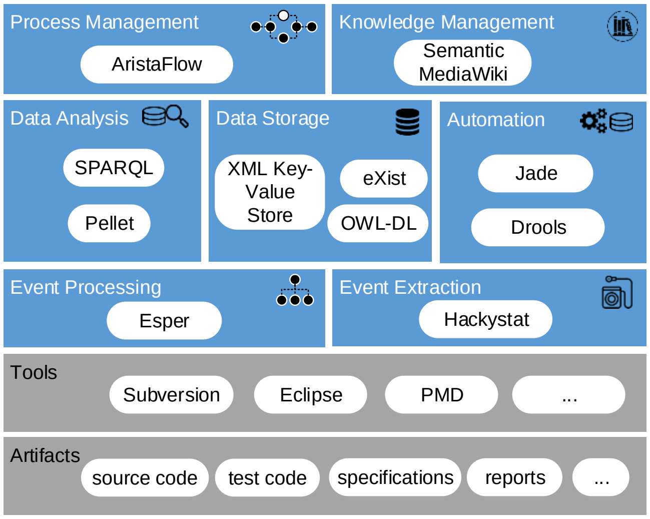

Figure 4. Framework Implementation for Software Engineering.

The framework also facilitates reasoning about higher level project information. Therefore, the framework integrates semantic web technology. This technology offers numerous advantages like advanced consistency checking or enhanced reuse possibilities among applications [23]. The data storage module therefore contains an OWL-DL (Web Ontology Language Description Logic) [24] ontology. That way, high-level project data can be stored in a standardized structured way. Moreover, ontologies provide capabilities for complex querying and the capability of reasoning about the contained data and inferring new facts. This is realized in the data analysis module with SPARQL [25] queries, SWRL [26] rule processing, and the reasoner Pellet [27]. This configuration fosters the integration of knowledge management with the high-level project information: The knowledge management component therefore integrates the Semantic MediaWiki [28]. Information entered in this wiki can be enhanced by machine-readable semantics enabling the framework to automatically access and distribute this information.

A framework aiming at holistic project support also needs components for automating as many tasks as possible. Therefor the automation module integrates the Jboss Drools [29] rules engine to execute simple automatisms, e.g., for converting reports. To support situations, in which goal conflicts arise, the framework also integrates the FIPA-compliant [30] multi-agent system (MAS) Jade [31]. Thus, it becomes possible to assign different goals to different autonomous agents that will pursue the respective goal.

To be able to support a software project holistically, its process and the various concrete workflows of the involved persons must also be managed in some way. To achieve this, the framework integrates the AristaFlow BPM suite [18][32] as process management module. That way, workflows can be composed out of existing services and human activities. The AristaFlow BPM suite guarantees correctness during modelling as well as enactment of the workflows. Furthermore, it has a feature crucial for process support in such a dynamic domain: Workflows can be adapted even when they are already running. Thus, their enactment is not tied to a pre-defined schema but can be tailored to the needs of the respective situation.

With this framework, we have implemented various use cases in order to achieve effective support of the involved persons. We will now provide details on one example of these being automatic provision of software quality measures. In that case, the framework automatically analyzed various reports regarding the quality of the source code and automatically selected matching software quality measures for existing problems. These measures were then automatically integrated into the developers' workflows in the best-matching situations. This was achieved by executing the following steps:

- Problem detection: To provide effective support, the awareness of existing problems is crucial. Therefore, the framework processes reports of external tools like PMD [33] for static source code analysis or Cobertura [34] for code coverage. Via rules processing the reports are transformed to a unified format and if defined thresholds for the contained metrics are exceeded, these are considered as problems and software quality measures are automatically assigned to them in the report. These are not the only problems that may exist but via the connections to various external tools using connectors and sensors the framework is aware of the execution of various tasks in the projects like testing or profiling and can detect their absence. The latter is also included in the problem report.
- Quality opportunity detection: To be able to distribute software quality measures automatically to concrete persons, the framework must be aware of their situation to not overburden them and provide quality measures matching their situation (e.g., the artifacts they are working on currently). This is enabled through the integration of the development process into the framework. Thus, the framework is aware of the planned activities and their timely planning as well as assignment to concrete persons. If a person finishes an activity early, the remaining time can be filled with a quality measure. As one cannot rely on people finishing their activities earlier as planned, there is also a quality overhead factor that allows for defining a certain percentage of the project to be reserved for quality activities.
- Measure tailoring: When the framework recognizes an opportunity for a quality activity, it triggers a measure proposal procedure. As a first step, the problems and assigned measures are strategically prioritized in line with the projects quality goals. This is achieved by an automated implementation of the Goal Question Metric (GQM) technique [35] realized with autonomous agents as illuistrated in Figure 5. Each agent pursues one quality goal like maintainability or reliability. Using the GQM structure, the agents can relate the metrics with violated thresholds to their respective goal. This is achieved by extending the standard GQM structure: Besides the goals, questions, and metrics the extended structure also incorporates measures and the agents. In addition, different levels of KPIs are integrated: The KPI aggregates the values of one or multiple metrics. The QKPI, which is assigned to a GQM question, aggregates the values of multiple KPIs. Finally, the GKPI that belongs to a certain goal aggregates multiple QKPIs. Utilizing these KPIs, each agent can calculate a concrete value representing the state of the goal it pursues. That way, the agents can autonomously prioritize concrete software quality measures. Each agent has a number of points he can distribute on the measures. For proactive measures, the agents use a competitive bidding process, in which each agent tries to bring measures relating to his goal to execution. For reactive measures, the agents utilize a cooperative voting process where the cumulated value of points spent by all agents on a measure is used for ranking the measures. After that, measures matching the respective person's situation must be selected to aid affective application of the measures. To find matching points in the various workflows a person processes, the latter are semantically annotated with extension points. These are points where a workflow can be extended by inserting new activities into it. Thus, specific properties of the persons, the measures, and the extension points can be matched to finally select the right measure for the right extension point of the right person.
- Measure application: At the end of the quality measure distribution the integration in the operational workflows of the respective persons must be done. This is achieved by the capabilities of the AristaFlow BPM suite: Workflow instances can be adapted during runtime even if their processing has already started. The process management module utilizes these capabilities to automatically and seamlessly integrate the measures into the selected workflows at the chosen extension points.
- Quality trend analysis: The final step of the procedure is the continuous analysis of the products' quality to assess the effectiveness of the applied quality measures. This is achieved by continuously analyzing reports from external tools. Thus, it can be determined if previously detected quality

problems disappear. Moreover, via the GQM structure the development of the quality goals can also be monitored.

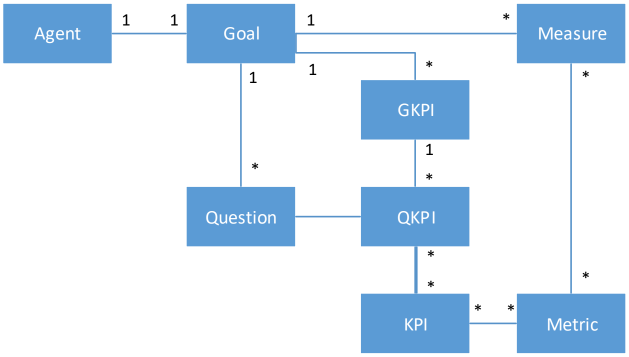

Figure 5. GQM Structure for Autonomous Agents.

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. In fact, two teams used the framework in a certain project and reported on its usability. Thus, we gained insights in the advantages the framework could enable in real usage. The main advantage was the support of better software quality. With features like automated software quality measure distribution or activity coordination many aspects that would have been forgotten by humans could be automatically supported which can lead to better quality source code and less bugs.

The second scenario involves a business use case in which different companies in a supply chain have to exchange sustainability information regarding their production [13]. 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 [36] as illustrated in Figure 6.

In this case, the focus was different than the one described in the software engineering domain: The target was not holistic support of entire projects with various use cases but the support of one complex use case. The crucial components were the generic connection to various external tools in the supply chain to obtain the contextual properties influencing the data collection process and the automatic generation of the latter by analyzing the properties. Therefore, a more tailored-down implementation of our framework idea was suitable.

The connection to the frameworks environment was realized by a set of connectors and adapters. Thus, it was possible to gather information from various external tools and provide the frameworks functionality to others. The latter was enabled by exposing a Java Service Provider Interface (SPI). Data collection was realized by web service adapters to use data sources that provided web services. In addition to this, specific connectors were created for the most prevalent in-house solutions in this domain. With the different connectors the utilization of information from prevalent tools of this domain, like BOMcheck [37] or IMDS [38], was possible. Knowledge management was also realized in a less automated fashion for this use case with a wiki and a bulletin board to support users in storing and retrieving information regarding the sustainable supply chain communication.

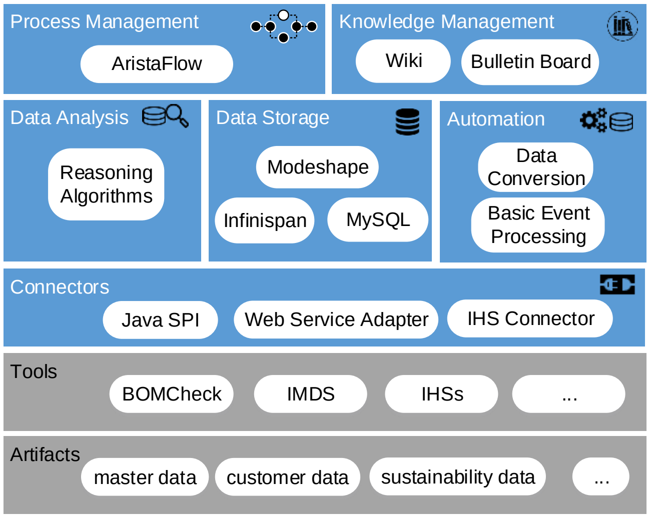

Figure 6. Framework Implementation for Supply Chain Data Collection.

The core of the framework was made up by the data storage, automation, and data analytics modules. Many different data sets had to be integrated as, e.g., master data, customer data, sustainability data and contents like reports. Furthermore, the ability to scale and provide high performance was crucial. Therefore, we opted for a combination of MySQL, Modeshape [39], and Infinispan [40]. The structured data with high focus on consistency was stored in MySQL while all relevant contents like documents and reports were stored in a content repository realized by Modeshape. For a performance increase we used the latter together with Infinispan, which acted as distributed cache. In the automation component, we implemented various data transformations to transfer the data obtained from external sources in a format our framework can process. Besides that, basic event processing was also integrated to automatically react to events in the changing environment. One example for this is the triggering of activities in case a certification of a customer was no longer valid, e.g., in case of changes to regulations.

For the data analysis module, we implemented reasoning algorithms that examine the obtained data from OEMs, regulations, the concrete requests, and suppliers. From this data, concrete context properties were extracted. By analyzing these properties and using the results to adapt processes, we were able to automatically create customized data exchange processes suiting different situations. For process management we opted for the AristaFlow BPM suite due to its capabilities for dynamic and correct process adaptation. In particular, we extended the process specifications with various properties to be matched with the context properties. To be able to build customized data exchange processes, we defined process fragments that were automatically composed to processes by our reasoning algorithms. Thus, it was not only possible to tailor these processes exactly to the respective situations but also to dynamically adapt long-running processes to changing situations. 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 [6][12]. 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) [41][42]. 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 [43] and model-driven approaches [44]. Hence, these frameworks could not provide holistic support for entire projects.

Another area comparable to the current approach for supporting knowledge-intensive processes are tools and frameworks enabling the technical enactment of flexible processes like Provop [45], WASA2 [46], Worklets [47], DECLARE [48], Agentwork [49], Alaska [50], Pockets of Flexibility (PoF) [51], and ProCycle [52]. Provop provides an approach for modeling and configuration of process variants. WASA2 constitutes an example of adaptive process management systems. It enables dynamic process changes at the process type as well as the process instance level. Worklets feature the capability of binding sub-process fragments or services to activities at run-time, thus not enforcing concrete binding at design time. DECLARE, in turn, provides a constraint-based model that enables any sequencing of activities at run-time as long as no constraint is violated. Similarly, Alaska allows users to execute and complete declarative workflows. Pockets of Flexibility is a combination of predefined process models and constraint-based declarative modeling. Agentwork features automatic process adaptations utilizing predefined but flexible process models. Finally, ProCycle provides integrated and seamless process life cycle support enabling different kinds of flexibility support along the various lifecycle stages. As a matter of fact, all of these approaches enable the flexible technical enactment of processes which constitutes a crucial feature when trying to support knowledge-intensive processes. However, to achieve higher-level support, process changes must be applied automatically on account of current data. This requires components for automatic data analysis and automatic process changes which none of the mentioned approaches provide. Only Agentwork provides rudimentary capabilities for automation but lacks the general applicability of our approach.

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 [37], 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 domains were we have successfully implemented concrete frameworks based on our idea of the abstract framework. In the software engineering domain we have shown how to achieve holistic support and guidance for the involved persons encompassing entire projects. Therefore, we have implemented support for various complex use cases like automatic software quality management support, automated coordination, and knowledge management. The second scenario we presented relates to sustainable supply chain communication. We have shown how to implement a tailored-down version of the framework to support one complex use case spanning the whole supply chain: The recursive request of sustainability data from suppliers. To achieve this, we have analyzed various different data sets in order to customly and dynamically create data collection processes matching the properties of the respective situation.

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] G. Grambow, “Utilizing Data Analytics to Support Process Implementation in Knowledge-intensive Domains,” Proc. 7th Int'l Conf. on Data Analytics (DATA ANAYLTICS 2018), pp. 1-6, 2018.

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

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

[4] C. Di Ciccio, A. Marrella, and A. Russo, “Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches.” J. Data Semantics 4(1), pp. 29-57, 2015

[5] R. Vaculin, R. Hull, T. Heath, C. Cochran, A. Nigam, and P. Sukaviriya, “Declarative business artifact centric modeling of decision and knowledge intensive business processes,” 15th IEEE Int’l Conf. on Enterprise Distr. Object Computing (EDOC 2011), pp. 151-160, 2011

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

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

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

[9] 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.

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

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

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

[13] 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.

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

[15] 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.

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

[17] 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.

[18] 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.

[19] OSGi Alliance: https://www.osgi.org/. [retrieved 02, 2019]

[20] W. Meier, “eXist: An Open Source Native XML Database,” Web, Web-Services, and Database Systems, Springer, ,pp. 169-183, 2009.

[21] P. M. Johnson, “Requirement and Design Trade-offs in Hackystat: An In-Process Software Engineering Measurement and Analysis System,” Proc. of the First International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, 2007, pp. 81-90.

[22] Espertech: http://www.espertech.com/esper/. [retrieved 02, 2019]

[23] D. Gasevic, D. Djuric, and V. Devedzic, “Model driven architecture and ontology development.” Berlin: Springer-Verlag, 2006.

[24] World Wide Web Consortium, “OWL Web Ontology Language Semantics and Abstract Syntax,” (2004)

[25] E. Prud’hommeaux and A. Seaborne, “SPARQL Query Language for RDF,” W3C WD 4 October 2006.

[26] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean,. “SWRL: A semantic web rule language combining OWL and RuleML,” W3C Member Submission, 21, 79, 2004

[27] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL Reasoner,” J. Web Semantics, 2006.

[28] M. Krötzsch, D. Vrandecic, and M. Völkel: „Semantic mediawiki,“ Proc. Int'l Semantic Web Conference, pp. 935-942, 2006.

[29] P. Browne, “JBoss Drools Business Rules. Packt P. Browne. JBoss Drools Business Rules” Packt Publishing, 2009.

[30] P. D. O'Brien and R. C. Nicol, “FIPA — Towards a Standard for Software Agents”, BT Technology Journal, 16(3), pp. 51-59, 1998

[31] F. Bellifemine, A. Poggi, and G. Rimassa, “JADE - A FIPA-compliant Agent Framework,” Proc. 4th Intl. Conf. and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents. London, 1999.

[32] M. Reichert et al., “Enabling Poka-Yoke Workflows with the AristaFlow BPM Suite,” Proc. BPM'09 Demonstration Track, 2009

[33] T. Copeland, “PMD Applied,” Centennial Books, ISBN 0-9762214-1-1, 2005

[34] Cobertura, http://cobertura.github.io/cobertura/. [retrieved 02, 2019]

[35] V. Basili, G. Caldiera, and H. D. Rombach, “Goal Question Metric Approach,” Encycl. of Software Engineering, John Wiley & Sons, Inc., pp. 528-532, 1994.

[36] 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.

[37] BOMcheck: https://www.bomcheck.net. [retrieved 02, 2019]

[38] International Material Data System: https://www.mdsystem.com/imdsnt/startpage/index.jsp. [retrieved 02, 2019]

[39] Modeshape: http://modeshape.jboss.org/. [retrieved 02, 2019]

[40] Infinispan: http://infinispan.org/. [retrieved 02, 2019]

[41] S. Bandinelli, A. Fuggetta, C. Ghezzi, and 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.

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

[43] 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.

[44] F. A. Aleixo, M. A. Freire, and 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.

[45] A. Hallerbach, T. Bauer, and M. Reichert, “Capturing variability in business process models: the Provop approach,” J. Software Maintenance and Evolution: Research and Practice, 22(6 7), 2010, pp. 519-546

[46] M. Weske, “Flexible modeling and execution of workflow activities,” Proc. 31st Hawaii Int'l Conf. on System Sciences, 1998, pp. 713-722

[47] M. Adams, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst, “Worklets: A service-oriented implementation of dynamic flexibility in workflows,” On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, LNCS, 4275, 2006, pp. 291-308

[48] M. Pesic, H. Schonenberg, and W.M.P. van der Aalst, “Declare: Full support for loosely-structured processes,” Proc. 11th IEEE International Enterprise Distributed Object Computing Conference 2007, pp. 287-298

[49] R. Müller, U. Greiner, and E. Rahm, “AGENT WORK: a workflow system supporting rule-based workflow adaptation,” Data Knowlage Engineering, 51(2), 2004, pp. 223-256

[50] B. Weber, J. Pinggera, S. Zugal, and W. Wild, “Alaska Simulator Toolset for Conducting Controlled Experiments on Process Flexibility,” Proc. CAiSE'10 Forum, LNBIP, 72, 2011, pp. 205-221

[51] S. Sadiq, W. Sadiq, and M. Orlowska, “A framework for constraint specification and validation in flexible workflows,” Information Systems, 30(5), 2005, pp. 349-378

[52] B. Weber, M. Reichert, W. Wild, and S. Rinderle-Ma, “Providing integrated life cycle support in process-aware information systems,” Int'l J. Cooperative Information Systems (IJCIS), 18(1), 2009, pp. 115-165