Components for a SOA with ESB, BPM, and BRM

Decision Framework and architectural Details

Arne Koschel, Andreas Hausotter
University of Applied Sciences and Arts Hannover
Faculty IV - Business and Computer Science
Ricklinger Stadtweg 120, 30459 Hannover, Germany
Arne.Koschel@hs-hannover.de,
Andreas.Hausotter@hs-hannover.de

Malte Zuch, Johannes Busch, Juri Seewald
University of Applied Sciences and Arts Hannover
Faculty IV - Business and Computer Science
Ricklinger Stadtweg 120, 30459 Hannover, Germany
Malte.Zuch@hs-hannover.de, Johannes.Busch@stud.hs-hannover.de, Juri.Seewald@stud.hs-hannover.de

Abstract - **To keep their competitive edge, enterprises need to change their operational processes in a flexible and agile manner. A Service-oriented Architecture (SOA) may help to meet these needs. One key feature of a SOA is the externalization of business process logic. However, the logic process is often complex, hard to understand and difficult to adapt. This issue is due to a mingling of process and decision logic. In order to ensure flexibility and agility, decision logic should be moved to a separate service. In previous work, we provided a decision framework, which recommends an approach to actually realize such a "rule service" conceptually. We apply our framework in particular to the German insurance domain utilizing a standardized insurance process scenario. The paper presents the resulting SOA architecture, which has an Enterprise Service Bus (ESB), a Business Process Management (BPM) / Workflow Management System (WfMS), and a Business Rules Management System (BRMS) / rules engine as key components. As a key contribution here, we provide several internal architectural and implementation details including deployment and runtime views of our architecture.**

Keywords - Business Process Management (BPM); Business Rules Management (BRM); Business Rules Management System (BRMS); Enterprise Service Bus (ESB); Service-oriented Architecture (SOA).

I. INTRODUCTION

A. Motivation

Workflow Management Systems support companies in the management and execution of business processes [25]. Nowadays, the latest challenges for insurance companies such as the dynamic business environment and compliance with legal requirements highlight the need for business agility [2][24]. Business agility requires individual, quick, and flexible composition and adoption of business processes [9][10]. This can be done in the context of Business Process Management (BPM). As a result of the composition and adaption, the number of decisions may rise within the processes. Hence, the complexity of the business processes can lead to a lack of business agility [10]. Business rules provide an opportunity to reduce the complexity of the processes, whilst the complex decision logic is encapsulated. The necessary changes with respect to agility often relate to the complex decision logic and not to the process or business logic. Thus, the separation of decision logic and process logic on the modeling and implementation level is a useful approach to reduce complexity.

Comprehensive service-oriented approaches have the potential to create business agility [27]. Thus, a service-oriented architecture (SOA) can help to address challenges like the dynamic business environment. The service-oriented integration of BPM and Business Rules Management (BRM) provides potential to change business processes in an agile manner [11].

The results of interviews with experts of the insurance service sector emphasized that issue to choose an adequate approach to automate the execution of business rules within service-oriented architectures with respect to a missing decision support. Considering the dynamic business environment in the insurance services sector, the topics of the presented work are of potential value for several insurance companies (at least) in Germany [2][24].

B. Contribution

Our previous work [11] presents a decision framework, which recommends an approach to realize a "rule service" conceptually. It serves as the groundwork for the key contributions of this article, which are:

- The application of the decision framework to scenarios particularly suitable to the German insurance domain, but easily transferable to similar environments. Thus, while we do not claim to provide too much novelty from an overall generic SOA/BPM/BRM perspective, we do contribute a valuable case study including detailed design decisions. Thus, our work provides value, which is especially applicable to the German insurance domain in general.
- A resulting SOA, which has as key technical components an ESB, a BPM system/WfMS, a business activity monitoring (BAM), and a BRM system.
- The detailed design of our SOA includes four steps: (1) initial design, (2) design decisions, (3) product evaluation for key components of the architecture, and (4) a resulting final architecture.

- Moreover, our SOA is applied to a standardized insurance process application scenario (“Goodwill Process”) working within the overall architecture from the German “Versicherungs-anwendungsarchitektur (VAA) [28]”. The VAA is a set of standardized insurance processes, the “insurance application architecture”.
- This paper significantly extends our work from [1]. In particular, we provide here much more conceptual and technical depth as key additional contributions. This includes more internal architectural and implementation details, for example, a runtime view and a deployment view of our architecture.

Our work takes place within the context of the current research activities of the “Competence Center – Information Technology and Management” (CC ITM) [5]. The CC ITM is a cooperation between IT departments from German insurance companies and our faculty. The aim of this cooperation is knowledge transfer and the combination of scientific research with practical experiences.

The remainder of the paper is structured as follows: In Section II we present prior and related work. In the following, we first show the application scenario in Section III, then our initial architecture in Section IV, implementing design decisions in Section V, an evaluation of products in Section VI, and eventually the resulting target architecture of the system in Section VII. Section VIII finalizes the article with a conclusion and an outlook to future work.

II. PRIOR AND RELATED WORK

The concept of a complex software architecture is always influenced by several factors. For handling the variability of decisions between those factors, a quantitative evaluation method can reduce complexity. In a previous work of the CC ITM, different concepts and technologies were discussed with such a quantitative evaluation method [18][16]. Therefore, different factors have been specified to build up a decision framework for identifying suitable business rule execution approaches. Further on, potential application scenarios have been identified by the CC ITM and the collaboration partners.

As a result, the standardized insurance process application scenario (“Goodwill Process”) was selected. The scenario, introduced in Section III (cf. [34][18] for an extended version), is inspired by a common insurance application architecture (“GAA”) used by the German insurance industry [28].

The basis of this work is specified by our partners from the insurance domain as well as the usage of the GAA. Within this specific domain, we contribute a valuable case study including detailed design decisions. Thus, our work provides novelty and value, which is especially applicable to the German insurance domain in general although might well be transferable to domains with similar requirements as well.

The required elements, which are to be implemented with a rule-based approach, were determined within this scenario. In this regard, we identified the business rule set “goodwill adjustment”. An extraction process for business rules identification from business process models is mentioned in [20]. This process is useful, because business rules are often not explicitly included in the process models. A decision guideline for distinguishing between business process and business rule is presented in [22]. Requirements concerning business rules technologies are defined in [2][24]. The variables for determining suitable solutions for business rule implementation are illustrated in [23]. As a result of the literature review, the decision guideline, the requirements and the variables provide a contribution to the decision framework. Since no previous research allows a simple choosing of an adequate business rules execution approach this decision framework is the first to extend the current state of research through the linking of factors, indicators and business rules execution approaches. The determination of the specific business rules execution approach depends on the elements, which are to be implemented with a rule-based approach.

Concerning our project, the software architecture has to fit the demands of the insurance business. Requirements such as privacy and security protection of customer data excluded peer-to-peer (P2P) solutions despite the advantage of the high availability P2P solutions could offer. Thus, solutions with discrete data storage options and a higher reliability concerning requests were considered. Especially, the service-oriented approach with an agile business rules solution was identified as most fitting for the insurance sector. The combination of high cohesion and loose coupling increases the flexibility and maintainability of complex and highly distributed software architectures [12]. In particular, an ESB can fulfill the requirements of highly distributed SOAs [6].

The physical integration instead of a logical one was identified as the most fitting solution in [16], regarding the general SOA approach of the CC ITM project. Further on, reliable messaging and security aspects of a physical ESB are also supporting the general demands of the insurance business in terms of security. Because of the whole software architecture consisting of distributed software components, a cloud solution was determined as a potential extension for the current BPM solution [16]. The paper [7] highlights the benefits of migrating BPM solutions into the cloud, to fulfill the increasing future demand of adaptive solutions in a dynamic business environment.

As a result of all these findings, the following Section III presents our application scenario from the German insurance domain. With this standard scenario, we will analyze its underlying SOA-, BPM-, BAM-, and BRM based architecture.

III. APPLICATION SCENARIO

A special application scenario has been applied to evaluate the prototypical implementation, which we have been described in [11]. This scenario is depicted in Figure 1. The scenario is a sub-process of the overall process “claim processing”. This overall process implements a standardized insurance companies use case, namely “handle a goodwill request” from the German “Versicherungsanwendungsarchitektur [28]”.

A goodwill payment is a compensation voluntarily granted by the insurance company without any obligation. The company checks whether compensation should be provided and - if so - determines its amount. The triggering event is the repudiation of cover. Its goal is to preserve the business relationship with the partner (customer). The task “Set goodwill adjustment” determines the goodwill amount and is a typical case for a business rule in the German insurance domain.

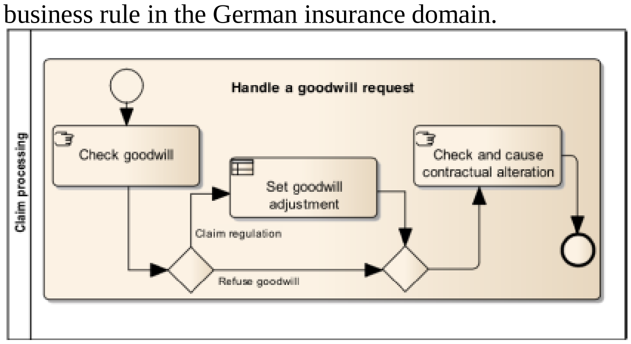

Figure 1: The goodwill process.

Within a process it makes sense to introduce certain tasks as business rules to gain flexibility or better maintainability. For example, some conditions at a certain decision point can change significantly more often than the overall business process. So, a flexibly changeable rule might offer more agility. These different requirements in flexibility directly influence the technically decisions about the actual rule realization/implementation.

IV. INITIAL ARCHITECTURE OF THE SYSTEM

To address these different requirements in flexibility regarding the implementation of business rules, we presented a decision framework in our prior work [11]. This paper compared different technical approaches for business rule execution (inference machine, database, configuration file and business applications) based on certain factors and indicators. By applying this decision framework, we decided to use a dedicated BRM inference machine for rule execution [11][16]. This approach has been identified as the most flexible one, especially with respect to the implementation of complex rules and larger rule sets. Next we will show the initial architecture of the system and will discuss its individual architectural components in detail. This architecture will process the described standard “Goodwill process” from Section III.

The architecture of the system for the “Goodwill Process” was composed from the following components (cf. Figure 2):

- The **WfM-Engine** (workflow management engine) was and still is the core of the whole architecture. It manages the business process, verifies the execution order of activities and routes the information flow between client and back-end. It contains a logical ESB to orchestrate different services. As concrete WfM-Engine, we use the product “Infinity Process Platform” (IPP) provided by SunGuard [13].
- The **BAMS** (business activity monitoring system) composes stored procedures and triggers. It is placed in an Oracle database and monitors the executions of activities used by the business processes. The BAMS is similar to complex event processing systems but uses a special form of logging. The monitoring allows the evaluation of business rule executions to improve the processes. It was designed by the CC ITM project.
- The **Client** component provides the functionality of the system to the user. It is currently a command console and menu based application, which allows the activation and execution of processes in the system. Currently, this client component is developed by the CC ITM and supports only the goodwill-scenario in a console-based user interface.
- The **Applications** provide the automatic execution of an activity or a single task. When they are invoked by the WfM-Engine, they execute the business logic that underlies the corresponding activity and report their result to the engine. Because the WfM-Engine only supports Enterprise JavaBeans (EJB) 2.x- and Web service calls, all applications are implemented as EJB 2.1 Beans.

Other components in Figure 2 are just supporting management and development background tools, which do not require a more detailed explanation here. The overall design decision and the composition of all its components to fulfill the requirements in flexibility and maintainability is described next.

V. DESIGN DECISIONS

The general big-picture of a software project is always the sum of every single design decision. The single decisions have to be chosen carefully. Therefore, a quantitative evaluation will help to support the decision-making.

A. Business activity monitoring system and business rules management system

To further improve the SOA aspect of the design, the introduction of a BRMS was considered. The BRMS would be responsible for managing the business logic and would also reduce some workload of the WfMS. Moreover, the BRMS would “user friendly” support modifications of business rules.

Overall, the BRMS is another “active rule” system, technical similar to the BAMS but with a different application-oriented purpose within our architecture. Regarding the responsibilities of these two systems, the introduction of a redundant component makes sense; the

BAMS has to monitor the whole WfMS architecture, while the BRMS only takes care of domain specific rules. This means, there will be two similar components for completely different tasks, combining both responsibilities would intermingle rules concerning different domains - a perilous path to take.

B. Enterprise service bus

The logical ESB, provided by the WfMS in use (IPP), was very restrictive in terms of supported applications and is not adaptable to offer security and transport protocols. Therefore, the advantages and disadvantages of a “homemade” physical ESB were compared with the advantages of the already existing logical ESB.

As presented in Table I, the advantages of using a physical ESB are significant and are outweighing the disadvantages.

Replacing the logical ESB with a physical one results in a more flexible architecture supporting the approach of loose couplings within the SOA. Therefore, the CC ITM team evaluated several solutions. For the concrete evaluation of the advantages and disadvantages of the different products, we will next discuss this evaluation with a quantitative approach.

VI. EVALUATION OF PRODUCTS

To find the best fitting set of products, a list of requirements was created and research was conducted on available alternatives. The evaluation process and its results are presented in this section.

Table I: ADVANTAGES AND DISADVANTAGES OF A LOGICAL AND PHYSICAL ESB FOR THE ARCHITECTURE.

	Advantages	Disadvantages
Logical	Already provided by the WfM-Engine	Dependence on the IPP WfM-Engine
	Responsibility for availability, security and reliability of the ESB outsourced to a third party company	Not expandable/modable for security, transport protocols and new interfaces
Physical	Lose coupling from IPP	More implementation work
	Less dependence on proprietary software components	Integration difficult or impossible due to the lack of knowledge
	More control and knowledge in self-programmed software components	
	More flexible software for security mechanisms, transport mechanisms and more interfaces	
	Future replacement of WfM-Engine easier (for example Stardust)	

A. ESB products

There are a lot of ESB solutions available today, analyzed in different publications. In this project, a long-list of possible ESB solutions was derived based on the publications [26][30][8][29][21][19]. These were the solutions:

Table II: THE LONG LIST OF ESB SOLUTIONS.

Apache Server Mix
Mule ESB
BEA System Aqualogic Service Bus
IBM WebSphere
Cape Clear
Oracle ESB
Fuse ESB
OpenESB
Talend Open Studio for ESB

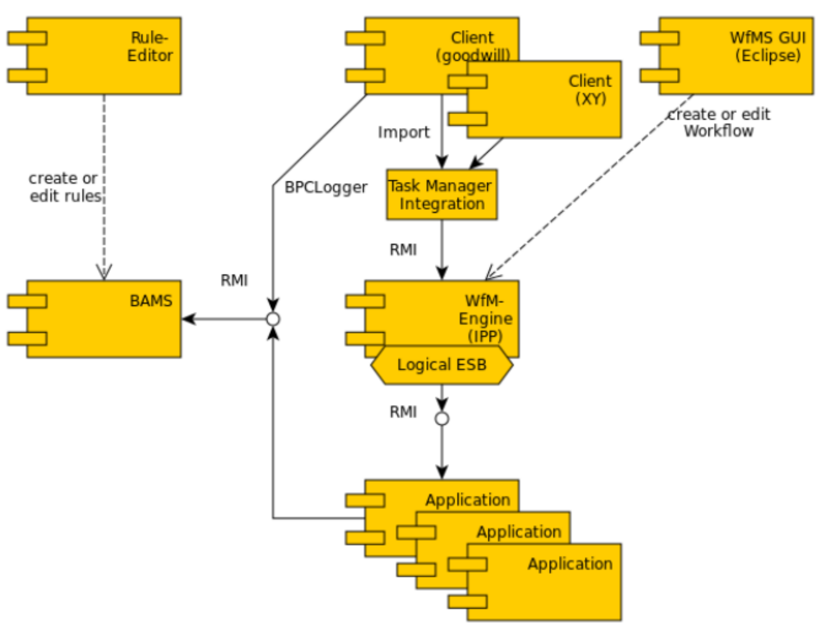

Figure 2: Initial architecture of the CC ITM project.

Because of some sources not being up-to-date, an additional study had to be undertaken by the project team in order to map the long list to the current situation on the market. It was discovered that some of the ESBs are included in different solutions now, because the manufacturers were acquired by other companies. For example, Fuse ESB currently belongs to JBoss Solutions from Red Hat Inc. [31].

In the next step, the long list was transformed into a short list. The conclusion drawn in [26][30][8][29][21][19] account for the choices made at this stage. Furthermore, special project requirements were used to extend the short list, such as the solution must:

- be open source and state-of-the-art
- work with both Windows and Linux operating systems respectively
- have an active support community
- provide an Enterprise JavaBeans connector for integration with existing components

The resulting short list is:

Table III: SHORT LIST OF ESB SOLUTIONS & EVALUATION (MAX. 100%).

Red Hat JBoss ESB	83.99%
Mule ESB	75.29%
OpenESB	83.98%
Talend ESB	89.87%

A full installation of each of the ESBs in the short list was not undertaken. Instead, the results of the comparative analysis in [3] were used, which describe detailed testing of solutions on different platforms, amongst which are Mule ESB, JBoss ESB and OpenESB. Talend ESB was evaluated in an interview [17] with an employee of Talend, using the criteria from [3]. These criteria belong to three categories, such as ergonomics, processing and environment.

For the evaluation of the ESB solutions, a value benefit analysis was performed. For this purpose, metrics and ratings for the criteria, and weightings for the categories have been defined [3]. Based on these specifications and the evaluation of the respective ESB product, the score has been calculated, which reflects the degree of fulfilment relative to 100% [3][17].

The combined results from both sources are also depicted in Table III. However, these numbers alone do not constitute the best solution, since possible problems of this result must be considered. Talend ESB was evaluated in 2013. The others were compared six years in before, so additional features might have been added in this period of time. Thus, despite it not having the best score, JBoss ESB was chosen by the team, because of its good documentation, wide usage and ability to run on JBoss Application Server 5.1, which was successfully used in the project before.

The compatibility of the ESB to the existing application server led to a low-effort integration into the architecture. After the ESB had been deployed on the server, it was necessary to ensure that the applications are not called by the logical ESB of the workflow engine directly any longer. Instead, the logical ESB will access the JBoss ESB which will call the applications. Referring to this, the JBoss ESB must provide an Enterprise JavaBeans service for the workflow engine. Therefore, the FacadeBean was created and the definition of the business process was altered, so that this Bean is accessed by the workflow engine when needed. These changes to the architecture are depicted in the final architecture diagram in Figure 7.

B. Business rule execution approaches (BREA)

In order to choose a business rules execution approach, a requirements analysis was undertaken, both for general business rules execution approach requirements and special requirements determined by the project. The former is defined in the Business Rules Manifesto from the Business Rules Group [4] and includes portability and user friendliness of the rule editor. The project requires the business rules execution approach to be:

- open source
- compatible with Linux and Windows operating systems respectively
- integrable with the existing JBoss ESB
- capable of processing complex business rules
- well documented, supported and constantly updated

In 2012, the team conducted a research on BRMS available on the market and created a list of suitable solutions. In order to assess the features of systems, an evaluation of BRMS was undertaken based on a criteria catalog developed by the project team based on a "Basel III" scenario [15][33]. The evaluation resulted in the BRMS short list in Table IV. The "Basel III" scenario asks for two typical indicators used by the underlying insurance business:

- liquidity coverage factor (LCR)
- net stable funding ration (NSFR)

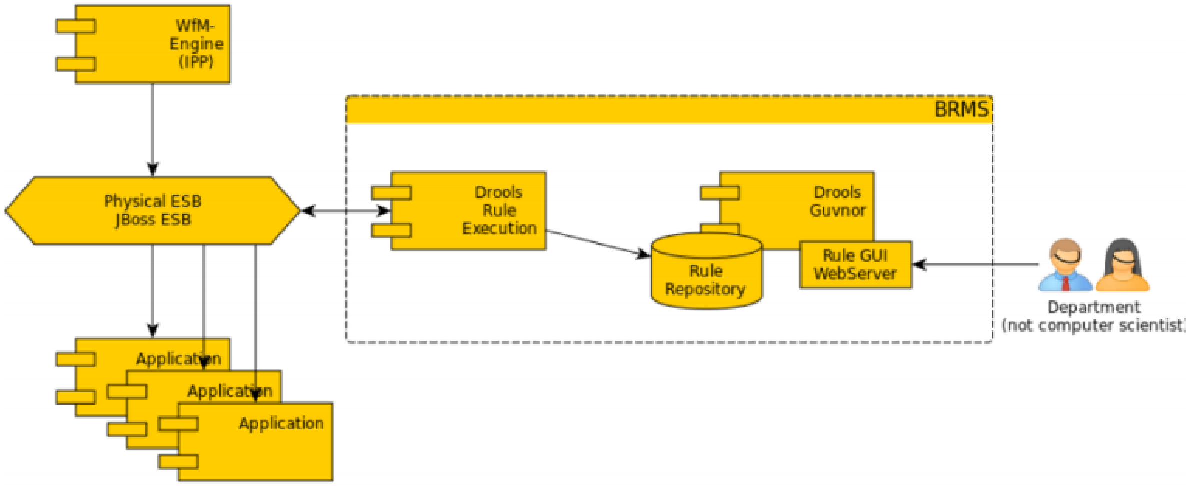

Figure 3: BRMS architecture with JBoss Drools and Guvnor.

In [15], the criteria are divided into nine groups: usability, ease of learning, run-time environment, performance, compatibility, functionality, safety and security, development and debugging, documentation. For each criterion, a score from one (worst case) to four (best case) was assigned to each product. The weighting of criteria was customized with respect to the specifics of the project. The evaluation results [14] are also presented in Table IV, although, those results cannot be used for judging about absolute quality of products. Nevertheless, due to restrictions of the project, an open source solution had to be chosen and therefore, JBoss Drools has been used in the prototype architecture.

JBoss Drools provides a complete system for business rules management, including a rule repository and a web server with a special site for rule management in Drools Guvnor. The BRMS architecture in the project is depicted in Figure 3.

Table IV: LIST OF BRMS SOLUTIONS& BRMS EVALUATION RESULTS (MAX. 100%).

Visual Rules	85.07%
JBoss Drools	61.09%
WebSphere ILOG JRules	77.19%

With this quantitative evaluation, we are able to identify the best fitting set of products to fulfill the requirements. The final system architecture build with those products is described next.

VII. TARGET ARCHITECTURE OF THE SYSTEM

There are several changes and optimizations between the basic and the target architecture of the system. Certain parts of the initial architecture have not been changed: Goodwill client, the connection between the client, the WfM-Engine and the BAMS are still as in the initial architecture. The original and modified parts are shown in Figure 7.

The first change of the initial architecture was the replacement of the logical ESB. For this purpose, JBoss ESB as a physical ESB was chosen as described before. Different applications will be called from the physical ESB instead of the logical one. Nevertheless, the logical ESB cannot be replaced completely, because it is an integrated part of the WfM-Engine. Also, it supports only EJBs and Web services connectors. Therefore, a connection between the logical ESB and physical ESB was developed. A simple Facade Bean represents this connection. It is called as an EJB from the logical enterprise service bus. All WfM-Engine calls will be channeled through the Facade Bean to different applications. Furthermore, the application calls in the process definition (XPDL file) were changed to leverage the physical ESB. To connect the applications to the JBoss ESB so called Services need to be described. The translation between the logical ESB and the Services is done by the Facade Bean. Moreover, the monitoring of these application calls is now handled by the JBoss ESB, for this purpose a connection between the JBoss ESB and the BAM-System was introduced.

The second change to the initial architecture was the integration of the BRMS. As stated before, the chosen BRMS is JBoss Drools. Its integration was realized through the connection between JBoss Drools and JBoss ESB. Furthermore, the BRMS was integrated into the process definition. Easily enough, a definition of a rule call is similar to an application call.

Management of the rule base is implemented by JBoss Drools Guvnor. Rules can be created or edited via a rule management website. Moreover, a rule storage (rule repository) is part of Guvnor.

The actual architecture is supported by decisions based on quantitative evaluation methods as well as on the expertise within the CC ITM team. We assume, a combination of quantitative methods and qualitative experience should offer an architectural design, able to challenge and be challenged by future demands.

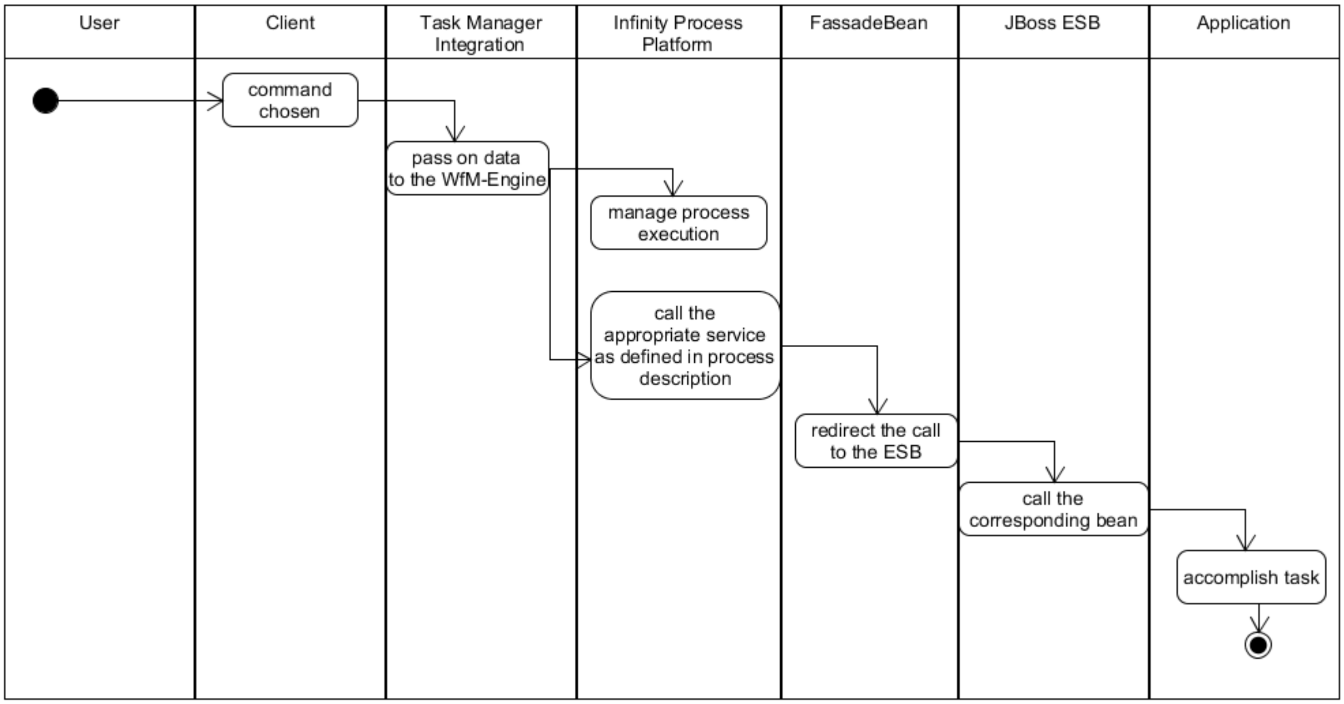

Figure 4: General control flow for user-initiated process tasks.

A. Runtime View

This section describes the dynamic view of the system's process [12]. The central process is "handle a goodwill request". It is a complex process that consists of many steps (service calls). Thus, the diagram of the process is not given here, but a short version is given in Figure 1. The general control flow for each step in the process initiated by the user is shown in Figure 4.

As mentioned before, the flow starts at the (command-line) client, passes the Task Manager and workflow engine. All service calls are mapped by the *FassadeBean* onto ESB service calls. Finally, the actual business logic is executed (service, application, etc.). All the most important user-initiated tasks follow this very similar pattern of calling the required activities and services to fulfil the desired task of the user. Following, the example use case "Select event of claim" (German: "Schadenfall auswählen") will be used to show the sequence of events in the system, based on the control flow from Figure 5.

The use case "Select event of claim" is the first step of the process. To initiate this use case, the user has to choose the specific option ("task 31: Select event of claim") on the (goodwill) console client and provide the id of the damage event. This information is transferred to the task manager integration component, to be more precise, to the component *CommandManager*. This component is responsible for providing this data to the workflow engine, using defined *TaskHandlingCommands*.

Simplified, the WfM-Engine (IPP) then receives the data and moves on to the next step in the process described in the specification, thus starting the use case "Select event of claim".

This use case requires interactions with further components (eventually services) of the system, defined in the process specification (XPDL file) as a call of a method in the *FassadeBean*. As already described in the previous section, the *FassadeBean* is a component (stateless session bean) that was built to enable the communication between the logical ESB of the WfM-Engine and the physical ESB (JBoss), because the engine does not support direct calls to an ESB.

So in this case, the purpose of the *FassadeBean* is to build the corresponding ESB call (JBoss Message objects, etc.) and provide it with the call information (especially the damage event id, process id, etc.) needed for the application scenario.

Simplified, the ESB calls the component (service) that is responsible for the business logic which is required for the service *damage case processing (DCP)*. Inside *DCP*, *damageManager* works the concrete method (*damageProcess*) to accomplish the task and returns the specific value object. Initially, this value object is required by the workflow engine and is returned by the *FassadeBean*.

Futhermore, the *damageManager* inside *DCP* calls the BPCLogger from BAMS, so that the execution of the task is monitored. For automated steps, the WfM-Engine calls the respective functions following the same control path via *FacadeBean* and ESB.

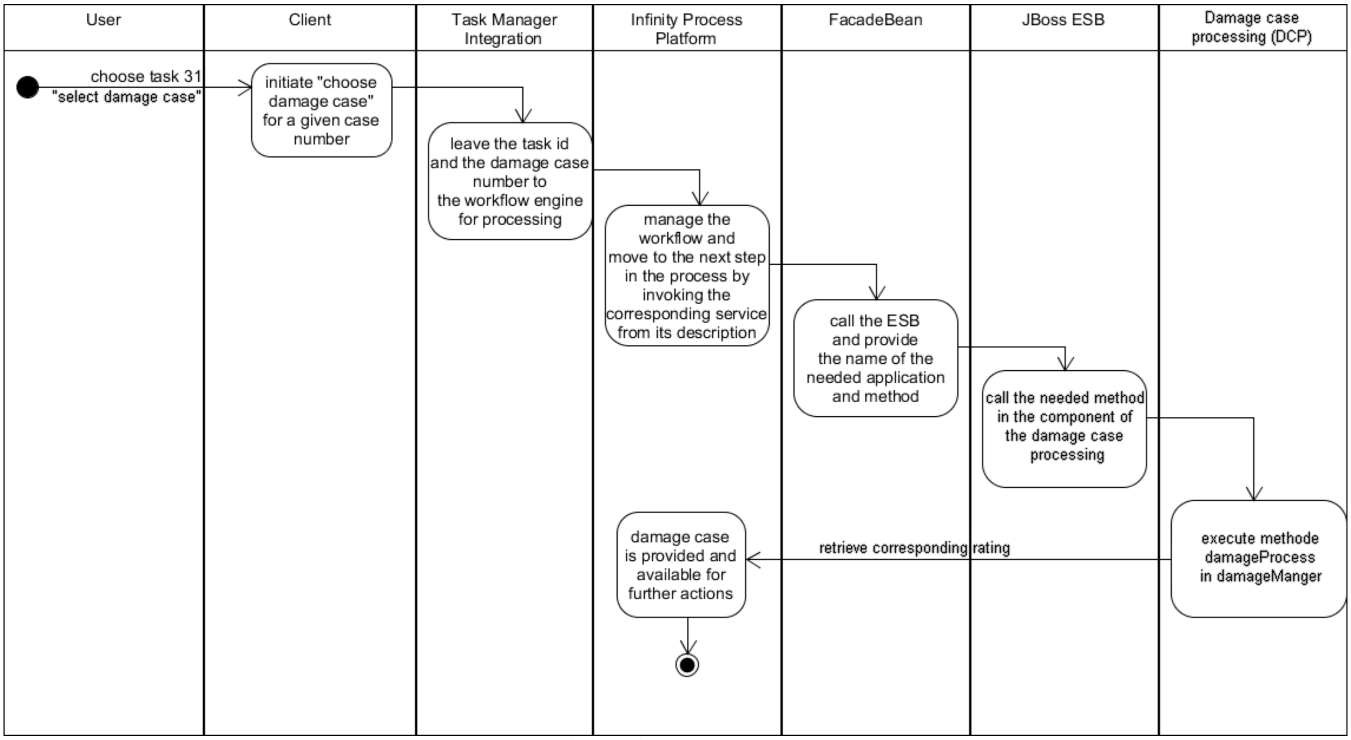

Figure 5: Sequence of actions for the use case "Provide damage event".

The Client gets updates about the status of the initiated task via the so called worklist through the task integration manager (*CommandMananger*). The update is shown on the console if needed.

The worklist itself is managed by the workflow engine. The information about the execution of tasks and processes is logged with BAMS and can be viewed in the audit trail database. Also, the WfM-Engine provides a web interface that can also be used for the same purpose. Whether a task has succeeded or not, is shown there. This web interface is also used for testing new created use cases. More information on testing is available in the corresponding programmer's guides of CC ITM.

After this presentation of the general runtime view, the following section will show the deployment view of the system.

B. Deployment View

As practical evaluation example, which provides even more technical depth, this section describes the technical implementation of a prototype of our implemented architecture [12]. For this purpose, we utilize the concept of a deployment view, which shows the actually implemented components, utilized technologies, technical communication protocols, and the deployment of all components to different nodes within a distributed system.

The deployment view of our architecture is depicted in Figure 6. The components that belong to the system are divided into the user client, the application server and the BAMS. Within this project, JBoss AS 5.1.0.GA is used as an application server. This application server also hosts the IPP 4.7.2 by executing the carnot.ear, as well as JBoss Enterprise Service Bus 4.1.2 and JBoss Drools 5.3. The hardware of the server, where the facade, workflow engine, ESB, Drools and the application scenario is deployed, realises the following hardware requirements:

- 1.5 GHz CPU
- 1 GByte RAM
- 70 GByte HDD

The usage of JBoss Drools / ESB is characterized by relatively low hardware demands. This allows to use a minimal and efficient hardware setup which already fulfils the demand in reactivity and processing speeds of the implementation of business rules and the execution of business processes. The BAMS component is deployed on an additional device. The device runs an Oracle Database Express Edition that realises the Business Activity Monitoring System. The hardware consists of:

- 1.5 GHz CPU
- 4 GByte RAM
- 40 GByte HDD

The BAMS is deployed on a separate device, because within the insurance business, there could occur high loads for the logging / monitoring. This high loads should not be able to affect the processing speed of the application server and its execution of business processes. So deploying the monitoring onto a separate device supports overall stability and reactivity of the overall system.

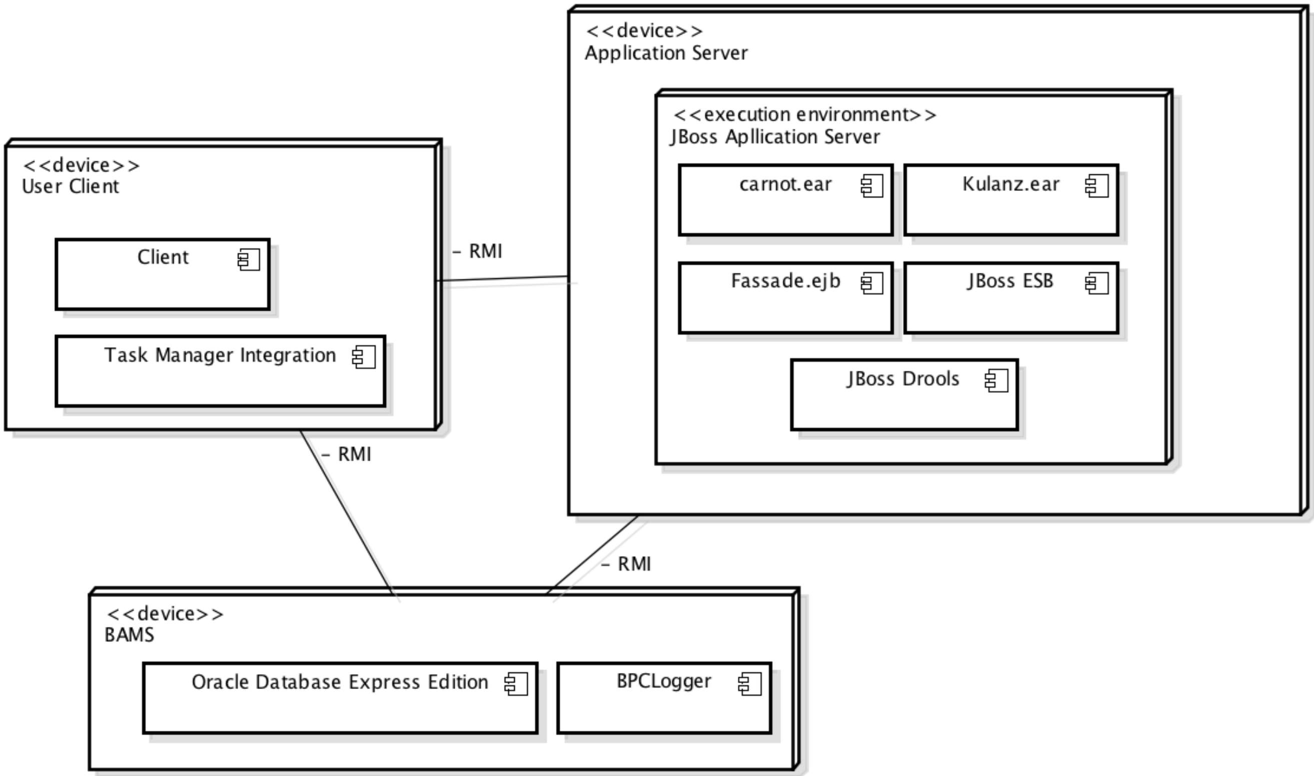

Figure 6: Deployment view of the system.

VIII. CONCLUSION AND FUTURE WORK

The presented overall architecture of the system consists out of four main components. This approach allows to process various client request and handle those in an efficient way, regarding complex business rules within the insurance business. As final summary, those for main components are characterized by the following:

- BAMS (business activity monitoring system): Used for flexible monitoring / logging, similar to complex event processing
- WfM (workflow management engine): The central component of the whole system, used for managing and executing all business processes.
- ESB (enterprise service bus): The (physical) ESB offers with loose couplings a high flexibility to handle interaction between various services within the insurance business scenario.
- BRMS (business rule management system): Especially the insurance business is characterized by specific rules. Those rules have to respect special (finance) laws and regulations. Therefore, the rule management have to be the most reliable and stable component of the system. The selected solution must fulfill high demand in trustiness and maintainability.

A. Conclusion

To manage the application landscape of businesses, for example, companies operating in the insurance services business, the combination of technologies such as SOA, business process management and business rules management is a promising approach. In order to ensure the optimum of agility and flexibility, the decision logic should be shifted to a separate SOA service.

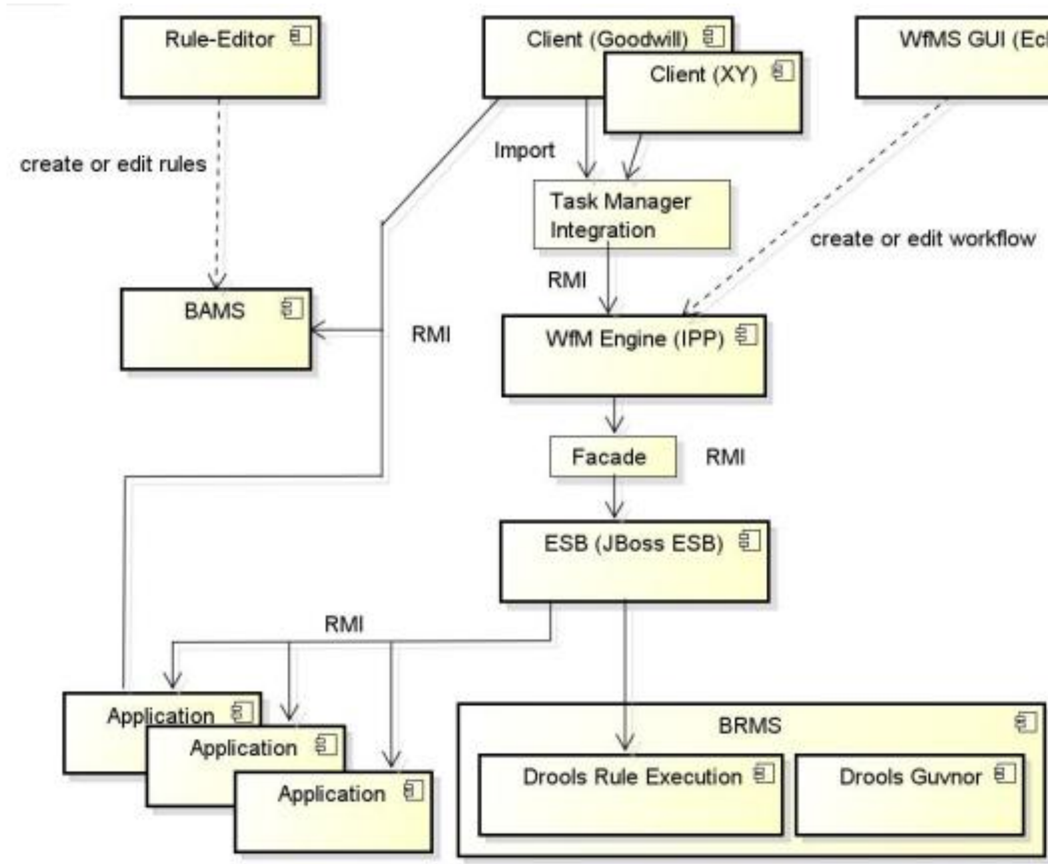

Figure 7: Final architecture of the overall system.

In our previous work, we presented a decision framework, which recommends an approach to realize such a "rule service".

As a key contribution of this article our decision framework is applied to a standardized insurance business process, namely "Handle a Goodwill Request". Starting from the initial design, making design and infrastructure decisions, we obtain an enhanced service-oriented target architecture with technical components such as ESB, BPM system, BAM, and BRM system.

Moreover, as a significant addition to our work from [1], we show in the present article much more details of the internal architecture an implementation. For example, we present the inner working dynamics of our architecture utilizing a runtime view of our architectural components as well as a deployment view of those components.

Thus, while we do not claim too much novelty for the general SOA/BPM/BRM case, we do provide a technically detailed SOA/BPM/BRM case study. This is in particular valuable for the German insurance industry [28] in general, but might well be transferable to domains with similar requirements as well.

B. Future Work

Based on these intermediate results, our subsequent research activities will focus on a detailed performance evaluation which may require a redesign of the target architecture. This evaluation is taking place within our current research project "QoS measurements for combined BRM, BPM and SOA environments in the insurance domain". As the insurance industry is receptive to cloud computing concepts and technologies - for example, product design and risk assessment frequently utilize cloud-based Software as a Service (SAAS) components - moving components of the target architecture towards the cloud might be a promising approach. Those components then might become generally usable, cloud-based services.

Therefore, the investigation of cloud computing solutions is another main activity of our research group. This includes conceptual and technical feasibility studies as well as security investigations and more. In our future work, we may also aim to address the aspect of "Threats to Validity" in some more depth.

Actually, the enhanced architecture contains some proprietary components, such as the BAM system. As businesses prefer to use standard infrastructure components, we intend to replace all proprietary components.

For this purpose, we (also) look at open source CEP tools. Another, but marginal issue is the optimization of the usability of the rule editor.

Last but not least our work on business rules management continues to take place within in our research.

REFERENCES

[1] A. Hausotter, A. Koschel, M. Zuch, J. Busch, A. Hödicke, R. Pump, J. Seewald, L. Varonina, "Applied SOA with ESB, BPM, and BRM – Architecture Enhancement by Using a Decision Framework," IARIA SERVICE COMPUTATION 2016: The Eighth International Conferences on Advanced Service Computing, Rome, Italy, 2016, pp 20-27.

[2] J. Boyer and H. Mili, “Agile Business Rule Development: Process, Architecture, and JRules Examples,” Springer, Berlin, Germany, 2011.

[3] A. Buchholz, R. Hohloch, and T. Rathgeber, “Vergleich von Open Source ESBs (Comparison of Open Source ESBs),” Institut für Architektur von Anwendungssystemen, Univerität Stuttgart, Stuttgart, Germany, Fachstudie Nr. 81, 2007.

[4] Business Rules Group. (2003, November 1). “The Business Rules Manifesto,” Version 2.0, edited by R.G. Ross) [Online]. Available: http://www.businessrulesgroup.org/brmanifesto.htm [retrieved: 02, 2016].

[5] Competence Center Information Technology and Management (CC ITM) [Online]. Available: http://f4.hs-hannover.de/forschung/kompetenzzentren/cc-itm/index.html [retrieved: 02, 2016].

[6] D. E. Chappell, “Enterprise Service Bus - Theory in Practice, O‘Reilly,” 2004.

[7] L. Chen, “Integrating Cloud Computing Services Using Enterprise Service Bus (ESB),” College of Business, Black Hills State University, 2012 [Online]. Available: http://sciedu.ca/journal/index.php/bmr/article/ view/674/387 [retrieved: 02, 2016].

[8] S. Desmet, B. Volckaert, S. Van Assche, D. Van Der Weken, B. Dhoedt, and F. De Turck, “Throughput Evaluation of Different Enterprise Service Bus Approaches,” in Software Engineering Research and Practice (SERP), Las Vegas, Nev., 2007, pp. 378-384.

[9] M. Dohring, L. Karg, E. Godehardt, and B. Zimmermann, “The Convergence of Workflows, Business Rules and Complex Events,” in 12th Intl. Conference on Enterprise Information Systems (ICEIS), 2010, pp. 338-343.

[10] T. van Eijndhoven, M. E. Iacob, and M. L. Ponisio, “Achieving Business Process Flexibility with Business Rules,” in 12th Intl. IEEE Enterprise Distributed Object Computing Conference (EDOC), 2008, pp. 95-104.

[11] C. Gath, A. Hödicke, S. Marth, J. Siedentopf, A. Hausotter, and A. Koschel, “Always Stay Agile! – Towards Serviceoriented Integration of Business Process and Business Rules Management,” in 6th Intl. Conf. on Advanced Service Computing (Service Computation), IARIA, Venice, Italy, 2014, pp. 40-43.

[12] M. Gharbi, A. Koschel, A. Rausch, and G. Starke, “Basiswissen für Softwarearchitekten (Fundamental Knowledge for Software Architects),” 2nd ed, dpunkt, Heidelberg, Germany, 2014.

[13] M. Gille, “The Benefits of Implementing Workflow,” [Online]. Available: https://www.sungard.com/~/media/fs/wealth-retirement/ resources/event-docs/Omni-Summit-Presentations/Implementing%20Workflow.ashx [retrieved: 02, 2016].

[14] M. Diekmann, C. Gath, J. Harms, V. Jartsev, B. Kleine, and J. Siedentopf, “Business Rules Management - Evaluierung von Technologien und Werkzeugen für das Business Rules Management (Business Rules Management – Evaluation of Technologies and Tools for Business Rules Management),” CC ITM, Technical Report, 2012.

[15] J. Harms and B. Kleine, “Fragenkatalog Business Rules Management (Questionnaire: Business Rules Management),” CC ITM, Technical Report, 2012.

[16] A. Hödicke, “Concept and Prototypical Implementation of a Serviceoriented Software Architecture for a WfM- and BRM-System,” M.S. thesis, Dept. Computer Science, Hochschule Hannover University of Applied Sciences and Arts, Hanover, Germany, 2013.

[17] A. Hödicke, “Evaluation of Talend ESB,” CC ITM, Technical Report, 2013.

[18] A. Hausotter, C. Kleiner, A. Koschel, D. Zhang, and H. Gehrken, “Always Stay Flexible! WfMS-independent Business Process Controlling in SOA,” in: IEEE EDOCW 2011: Workshops Proc. of the 15th IEEE Intl. Enterprise Distributed Object Computing Conference, IEEE: Helsinki, Finland, 2011, pp. 184-193.

[19] T. Krussman, A. Koschel, M. Murphy, A. Trenaman, and I. Astrova, “High Availability: Evaluating Open Source Enterprise Service Buses,” in Proc. ITI 2009 31st Intl. Conf. on Information Technology Interfaces, Cavtat/Dubrovnik, Croatia, 2009, pp. 615-620.

[20] O. Levina, O. Holschke, and J. Rake-Revelant, “Extracting Business Logic from Business Process Models,” in 2nd IEEE Int. Conference on Information Management and Engineering (ICIME), 2010, pp. 289-293.

[21] L. MacVittie (2006, March 10). “Review: ESB Suites,” Network Computing, CMP Media LLC [Online] Available: http://www.networkcomputing.com/wireless-infrastructure/review-esb-suites/d/d-id/1216317 [retrieved: 01, 2016].

[22] M. Zur Muehlen, M. Indulska, and K. Kittel, “Towards Integrated Modeling of Business Processes and Business Rules,” in 19th Australasian Conference on Information Systems (ACIS), 2008, pp. 690-697.

[23] M. L. Nelson, R. L. Rariden, and R. Sen, “A Lifecycle Approach towards Business Rules Management,” in 41st Hawaii Intl. Conf. on System Sciences (HICCS), 2008, pp. 113-123.

[24] M. Schacher and P. Grassle, “Agile Enterprises with Business Rules: The Business Rules Approach,” Springer, Berlin, Germany, 2006.

[25] T. Schal, “Workflow Management Systems for Process Organisations,” Springer, Berlin, Germany, 1996.

[26] Z. Siddiqui, A.H. Abdullah, and M.K. Khan, “Qualified Analysis b/w ESB(s) Using Analytical Hierarchy Process (AHP) Method,” in Intelligent Systems, Modelling and Simulation (ISMS), Kuala Lumpur, Malaysia, 2011, pp. 100-104.

[27] G. Starke and S. Tilkov (editors), “SOA Expertenwissen (SOA Expert Knowledge),” dpunkt, Heidelberg, Germany, 2007.

[28] GDV (Gesamtverband der Deutschen Versicherungswirtschaft e.V. – General Association of the German Insurance Industry), “Die Anwendungsarchitektur der Versicherungswirtschaft: Das Objektorientierte Fachliche Referenzmodell (The application architecture of the German insurance business – The functional object-oriented reference model,” http://www.gdv-online.de/vaa/vaafehtml/dokument/ofrm.pdf, VAA Final Edition. Version 2.0., 2001 [retrieved: 02, 2016].

[29] K. Vollmer and M. Gilpin, “The Forrester WaveTM: Enterprise Service Bus, Q2 2006,” Forrester Research, Cambridge, Mass., 2006.

[30] R. Woolley, “Enterprise Service Bus (ESB) Product Evaluation Comparisons,” Utah Department of Technology Services, Salt Lake City, Utah, 2006.

[31] Red Hat Inc., “Red Hat JBoss Fuse: Overview,” [Online]. Available: http://www.jboss.org/products/fuse/overview/ [retrieved: 02, 2016].

[32] Bank für Internationalen Zahlungsausgleich, “Basel III: Mindestliquiditätsquote und Instrumente zur Überwachung des Liquiditätsrisikos (Basel III: Minimal liquidity rate and instruments to supervise the liquidity risk,” Basler Ausschuss für Bankenaufsicht, 2013, pp. 2. [Online]. Available: http://www.bis.org/publ/bcbs238de.pdf [retrieved: 02, 2016].

[33] Bank für Internationalen Zahlungsausgleich, “Basel III: Strukturelle Liquiditätsquote (Basel III: Structural liquidity rate),” Basler Ausschuss für Bankenaufsicht, 2014, pp. 1. [Online]. Available: http://www.bis.org/publ/bcbs271de.pdf [retrieved: 02, 2016].

[34] T. Bergemann, A. Hausotter, and A. Koschel, “Keeping Workflow Enabled Enterprises Flexible: WfMS Abstraction and Advanced Task Management,” in 4th Int. Conference on Grid and Pervasive Computing Conference (GPC), 2009, pp. 19-26.