Application of Scenario-driven Role Engineering to the *MinaBASE* Process Knowledge Database

Daniel Kimmig*, Andreas Schmidt†*, Klaus Bittner*, and Markus Dickerhof*
*Institute for Applied Computer Science
Karlsruhe Institute of Technology
Karlsruhe, Germany
E-mail: {daniel.kimmig, andreas.schmidt, klaus.bittner, markus.dickerhof}@kit.edu
†Department of Informatics and Business Information Systems,
University of Applied Sciences, Karlsruhe
Karlsruhe, Germany
E-mail: andreas.schmidt@hs-karlsruhe.de

***Abstract*—Collaborative systems, which are often used in short-term virtual enterprises or long-term cooperation networks, often contain informations about the manufacturing and fabrication competences of the technology partners which should only be made available to a very restricted group of persons for example to support feasibility studies in the context of actual customer requests. This is a new important feature to be supported in nowadays knowledge management systems. Hence, the goal of this paper is so to present a methodology for implementing an access system that supports the definition of fine granular access rights for cooperative process knowledge management systems, that could protect such sensible information. In this paper we present an adaption of the scenario-driven role engineering method to the special needs in a collaborative process knowledge management system with fine granular access requirements. Beside the adaption of the scenario-driven role engineering method to such a system, the adapted method will be concretely applied to the process knowledge management system *MinaBASE*, which was developed in our institute. To complete, an implementation will be shown with the help of the inversion of control framework "Spring Security". Here static aspects as well as dynamic aspects of security will be presented. The paper shows in a detailed manner the usability of the scenario-driven role engineering method for applications in the field of collaborative knowledge management.**

***Keywords*-Data confidentiality and integrity; knowledge management; RBAC;**

I. INTRODUCTION

Both corporate groups as well as small and medium-sized enterprises (SME) are experiencing increasing competition and shorter product lifecycles. The resulting necessity of shortening the product development process also has to be mastered by enterprises in the field of microsystems technologies (MST) that are characterized by a high interdisciplinarity and complex, multi-stage, and hardly standardized fabrication processes. Frequently, every product is produced by an individually tailored fabrication process [1]. While larger MST enterprises still manage a wide spectrum of technologies, SME rather tend to offer solutions in a high specialized area. To offer more complex solutions, they establish technical partnerships with other SME. These may have the form of short-term virtual enterprises or long-term cooperation networks. To support such organization forms in the field of MST, the *MinaBASE* process knowledge database was developed by the Institute for Applied Computer Science of Karlsruhe Institute of Technology. By means of this database, the manufacturing and fabrication competences of the technology partners are made available to a central coordinator who then assesses technical and economic feasibility. This information, however, includes internal know-how, the confidentiality, secrecy, and integrity which is of crucial importance to the company's existence. Acceptance of *MinaBASE* therefore does not only depend on whether it meets functional requirements, but also on aspects like safety and access protection. To prevent an undesired disclosure of company secrets of a technology partner, access shall be controlled by the established role-based access control (RBAC) [2]. Here, authorizations are not assigned directly to subjects, i.e., the users of a system, but to abstract roles to which the users are assigned afterwards. In this way, the frequently error-prone maintenance expenditure is reduced and safety is increased. However, this requires the definition of an adequate role concept. Information systems often use standard models with system-wide administrators, owners of information objects, and guests having restricted read access. It is not considered in which business processes the system is used and which particular requirements result in terms of confidentiality and data integrity. For *MinaBASE*, it should be possible to provide a partner with insight in the product-related properties of a microsystem during the sales process, but not to disclose the configuration of machine parameters to produce these properties. Such an application exceeds the modeling capacities of the standard role concept described, as external partners are not the owners of this information. This results in a high expenditure to maintain the finely detailed authorizations. This problem will be solved by

 ISBN: 978-1-61208-010-9

a role concept individually adapted to *MinaBASE* . This role concept is based on a systematic approach, scenario-driven role engineering. The main contribution of the paper is twofold: First, the adaption of the scenario-driven role engineering method will be adapted to the requirements of collaborative knowledge management systems. And second, the suitability of existing access frameworks to implement the adapted method will be shown by means of the IoC-framework spring security.

The paper will be structured as follows: In the next section, the *MinaBASE* process knowledge database will be presented. Then, the scenario-driven role engineering approach [3] and the adaptations to the background and objective of *MinaBASE* will be outlined. Afterwards, the model will be applied and a role concept for RBAC ensuring data integrity and confidentiality for *MinaBASE* will be derived. The final section describes the implementation of this role concept within an IoC-Framework by demonstrating how Spring Security and technolgies such as AOP can be used to fulfill static and dynamic security requirements.

II. *MinaBASE* Process Knowledge Database

The knowledge required to produce values added is no public property, but a company resource that has to be administrated efficiently in order to ensure economic success. To support this process, knowledge management systems have been established [4]. In process-oriented knowledge management [5], these methods are applied to highly knowledge-intensive fabrication processes, as those used in MST. The *MinaBASE* process knowledge database was developed within the framework of the MicroWebFab joint project funded by the BMBF [5]. It was used by the technology partners for the structured storage of technical fabrication parameters of the methods and materials used in MST and of the partner-specific technical competences. In *MinaBASE* , the smallest information entity is the so-called technical aspect (TA). It is used to model materials, machines, and fabrication technologies [6]. By means of generalization hierarchies, TAs are arranged in taxonomies. The number and contents of taxonomy trees can be specified and modified during runtime, such that a flexible structure meeting MST requirements can be defined for the storage of fabrication know-how. TAs may be assigned properties referred to as technical parameters (TP). A TP is a string of characters, integers, or floating-point numbers in a certain unit and references an attribute, e.g., the density. The TP of a TA are inherited by lower partial hierarchies of the hierarchy tree in analogy to the object-oriented approach. Hence, no error-prone multiple input is required. In addition, lower hierarchy levels can further refine the inherited TP by specifying general value ranges. Typing of the TP places them in a certain context, such that a TP refers to a product or its fabrication and, hence, is either product-specific or fabrication-specific. Product-specific TP describe the properties of a microsystem, such as the depth of a groove reached by the fabrication process of milling. Fabrication-specific TP refer to the machine configuration needed to produce a specific product property. For modeling the capabilities of a technology partner, competences [7] are considered to be a set of various TA from disjunct hierarchy trees, which is illustrated in Figure 1. This figure schematically represents the competence “injection molding of a rib with polystyrene using the Arburg Allrounder machine” together with some TP. From the hierarchy trees of process, machine, material, and geometry element, the TAs are selected. These TAs are characterized by own TP, such as the injection pressure of the injection molding process. The combination of these TAs results in the competence that is reflected by other TP, such as the edge quality and surface roughness. Consequently, a competence is a type of view of a certain combination of TAs with properties in the form of TP that apply to this combination only, i.e., that characterize the competence in more detail. TAs can be used in several competences. They represent reusable, encapsulated, smallest information entities. An extension of the *MinaBASE* concept has been developed in order to reuse these information entities to allow process modeling of manufacturing sequences based on semantic technologies [8].

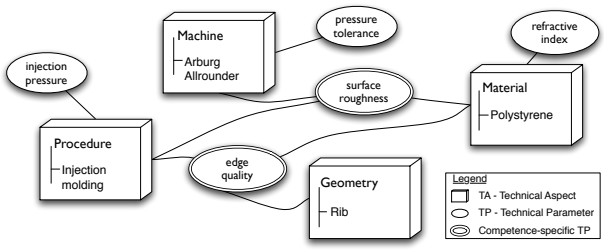

Figure 1. Schematic representation of a *MinaBASE* competence [8]

III. Scenario-driven Role Engineering

The term of role engineering (RE) in the context of RBAC means the design and specification of roles, authorizations, secondary conditions, and restrictions as well as of a hierarchic role model [9]. RE is used to create a concrete model for RBAC-based access control. In [3], Scenario-driven role engineering (SDRE) is defined as an approach based on scenarios, such as sequences of actions and events from the user’s perspectives. This sequence in a scenario can be subdivided into subscenarios and steps. Scenarios are embedded in a task, i.e., a problem or a work area, which links the scenarios of a system with its users. The users mostly apply a system to fulfill a task of their work profile or their job description. This structurization into various levels serves to break down a job description of a user into atomic steps, each of which may be associated with an authorization to access a resource. For various types

 ISBN: 978-1-61208-010-9

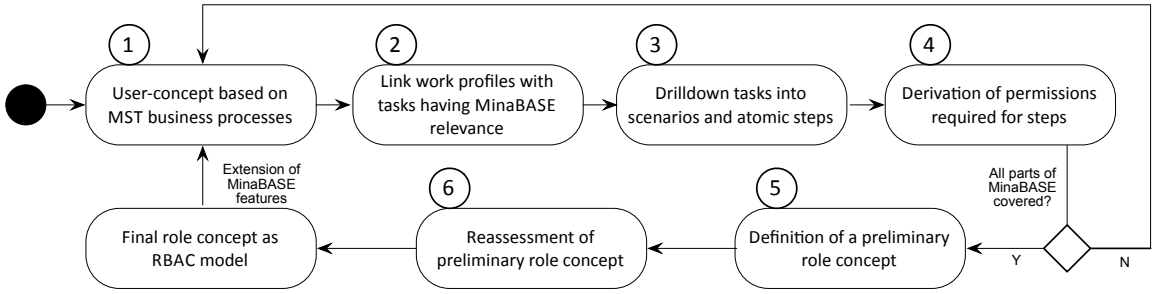

Figure 2. Scenario-driven role engineering according to [3] with adaptions

of users, the minimum amount of authorizations required for the execution of the tasks can be derived. In this way, the principle of least privilege [10] is implemented. For documentation, various models are generated by the SDRE approach, which are interlinked in terms of contents and used for the derivation of the role concept. The *scenario model* describes all scenarios and steps, *task definitions* serve to structure scenarios, the *work profile* summarizes tasks for job descriptions. The *permission catalog* lists the individual permissions or authorizations. It may be complemented by a constraint catalog of special limitations [3]. While the permission catalog is focused on static assignments of authorizations to specific resources, constraints describe dynamic conditions, which are evaluated at runtime. Hereinafter, the SDRE process will be described in general. First, the use scenarios of the system are compiled and their actions and events are documented. Then, subscenarios and steps are defined and the authorizations required for them are included in the permission catalog or special limitations are listed in the constraint catalog. When this step is completed for all scenarios, similar scenarios are generalized. Very complex scenarios are divided into smaller parts which are then included in the scenario model. On this basis, tasks are formed by grouping scenarios. These tasks are then classified into various work profiles. This results in a preliminary role concept and minimum authorizations can be assigned to the individual activities. As a rule, this preliminary model contains duplicates of roles with identical authorizations, which then have to be fused in a last step. This yields the RBAC model as a role concept. The SDRE process represents a systematic approach to RE. It was applied to information systems for the health care sector by the technical committee of HL7 already [11]. Due to this practical test, SDRE in principle may be applied to *MinaBASE* . However, certain adaptations are required, because the background and objective of *MinaBASE* differ from those of the HL7 systems. The paramount objective of *MinaBASE* is the support of knowledge-intensive business processes of MST enterprises by a structurization of the knowledge required for the execution of these business processes. A criterion for the acceptance of knowledge management is its integration in workflows of the users and an efficient and complete coverage of information needs [12]. As such the SDRE approach is to be applied to the use of *MinaBASE* in business processes of MST enterprises and cooperation networks. The model given in [3] is therefore subjected to the following adaptations:

- In the standard SDRE methodology, scenarios for a system are the main input, to which required authorizations are allocated. Afterwards these scenarios are generalized and assigned to tasks and work-profiles which create a preliminary role concept, that needs to be revised afterwards.
  For *MinaBASE* however an alternative input is more persuasive. Instead of starting with the scenarios of the system, work areas within the business processes of an MST-company are examined, whether they include tasks in which *MinaBASE* can be used to increase added value. To these tasks scenarios will be assigned in order to obtain the information, which resources are required for fulfilling them and what authorizations are needed. Based on this information, a preliminary role concept can be derived analogous to the standard model due to the minimal set of authorizations for each role. By these adapations, a switch of input variables to the methodology - the basic principle of SDRE is preserved, while better results for the creation of the role concept are expected, because of the adapted methodology being closer to the business processes of a MST-company.
- The scenarios to be formulated are not based on consequences of actions and events, but will also contain all definable steps. Although these do not occur in sequential order, they can be characterized by a certain access authorization.
- For reasons of clarity, special limitations extending beyond the static allocation of authorizations are included directly in the permission catalog and not in the

constraint catalog, such that both models fuse.

The adapted process is illustrated in Figure 2. It comprises six steps, the execution of which shall be described in more detail in the following section.

IV. APPLICATION OF SDRE TO *MinaBASE*

Using parts of the models created by the SDRE process, it shall now be demonstrated how the role concept can be generated systematically.

A. Step 1: Generation of the User Concept

Application of the model is based on an analysis of the business processes of a model MST company for possibilities of using *MinaBASE* and for activities, where the use of *MinaBASE* may result in a value added. Functions and units of an MST company, which may be potential users of *MinaBASE*, are:

- *Sales, external guest: MinaBASE* supports the sales process in the strategic assessment of the feasibility of customer orders, because these decisions can be made based on an IT documentation of competences and fabrication know-how. Strategic means that a general decision is made without taking into account technical details. In addition, the customer order is typed depending on whether a standard product is to be manufactured or a specification has to be met by enforcing the development in a project. The documented competences can be used as a database for sales promotion. External guests, e.g., customers or suppliers, may be given access to the system in order to inform themselves about fabrication processes used by the company or the cooperation network.
- *Project management, development:* If a customer order is classified to be not directly producible by the sales division, a project team is established based on the customer's specification. This team is composed of the project manager and technical experts. In an iterative process, they specify general solution alternatives, the commercial feasibility of which is assessed. In addition, solution approaches, such as functional patterns or prototypes, are developed in detail, the technical feasibility of which is guaranteed. Upon successful agreement with the customer, exact fabrication planning is started in the next step. Planning is based on the results of the development of a commercially and technically feasible solution.
- *Construction, fabrication planning:* Planning of fabrication, i.e., of the individual steps of production flow, may be initiated by a successful development process or a directly producible customer order, e.g., the repetition of an already executed fabrication process. In the latter case, *MinaBASE*, a system for process-oriented knowledge management, provides support by the storage of process elements of process steps and process sections and their combination in process chains, as this allows for the direct use of already executed fabrication processes [8]. This principle in weaker form may also be applied to fabrication planning based on a technical solution alternative from development. By copying or adapting existing process models or process elements, planning of the fabrication process can be accelerated. Construction and fabrication planning result in a detailed schedule for production and defined quality management tests, during which data are measured in the production process.
- *Quality management, production:* Production focuses on the execution of the process steps defined by fabrication planning in a process chain to execute the order placed by the customer. Technicians working at the machines have direct access to production and are capable of using technical parameters of the individual process elements of the process chain for adjusting the machine parameters and of measuring real data during the tests. Various areas of quality management are covered. New fabrication knowledge of attributes and parameters of process elements is generated.

Work-Profile	Task
Project management (PM)	Compose group of technical experts
	Coordination of orders and manufacturing sequences across boundaries of organizational units regarding process dependencies
	View competences of organizational units
	Analyze process-chains of organizational units
	[…]

Figure 3. Work area project management with associated tasks from the work profile model

B. Step 2: Definition of Work Profiles and Task Definitions

According to the adaptations to the SDRE model, *MinaBASE* tasks are assigned to the enterprise units or work areas listed in the previous section. Figure 3 shows a part of the work profile model. In the work area of project management for the iterative development of solutions for a not yet solved development problem of a customer order, tasks are identified, to which the *MinaBASE* resources can be applied. These tasks are the pooling of technical experts, the analysis of fabrication competences and process chains, and the coordination of process dependencies beyond organizational units. The complete work profile model contains all tasks to which *MinaBASE* may be applied. These are the input variables for the detailed assignment of scenarios, steps, and authorizations to access resources in the following step.

 ISBN: 978-1-61208-010-9

Task	Actor	Scenario/Step	Op	Resource	Permission	Constraint
Compose group of technical experts	PM	View organizational units	R	Enterprise Table	{PL, R, Enterprise_Tab}	-
	PM	Inquire contacts (work scheduler)	R	Enterprise-User Table	{PL, R, Enterprise-User_Tab}	-
	PM	View competences of organizations	R	Competence-Enterprise Table	{PL, R, Enterprise-Compet_Tab}	-
Coordination of orders and manufacturing sequences across boundaries of organizational units regarding process dependencies	PM	View orders	R	Order_Table	{PL, R, Order_Tab}	-
	PM	View associated organizations of specific orders	R	Order –Enterprise Table	{PL, R, Order –Enterprise Tab}	-
	PM	View associated manufacturing competences of orders	R	Order-Competence Table	{PL, R, Order-Competence_Tab}	MR_OE-Filter
	PM	View manufacturing sequence and precess dependencies as attribute values	R	Order-Competence-Attribute/Values Table	{PL, R, Order-Competence-Attribute/Values_Tab}	MR_OE-Filter

Figure 4. Refinement of tasks in use scenarios and assignment of authorizations to access the resources needed

C. Steps 3/4: Refinement of Scenarios and Assignment of Authorizations

For the first two tasks mentioned in the previous section, namely, the pooling of experts and the coordination of process dependencies, a part of the fused permission and constraint catalog is illustrated in Figure 4. For every scenario or every step, the associated operation on an object or resource is modeled, with R denoting read access (read), C denoting the creation of a new entry (create), U meaning processing (update), and D the deletion (delete) of a resource. The information of which actor accesses which resource with which operation is encapsulated as a permission by a triple of the type (actor, operation, object). At last, limitations or constraints of access are specified. For the first task, the organizational units, contact data of technical experts, and competences of the organizations stored in *MinaBASE* are considered as use scenarios. Read access (R) to the tables of the database and application components is required. In analogy, the second task is handled. Here, order data and detailed, production-related attribute values of process dependences are needed. In addition, an entry in the constraint catalog is made to ensure that the actor sees only those attribute values that are characterized as project-specific property and not as production-specific, internal know-how of an organization. This constraint cannot be implemented as a static authorization, as the assignment is made dynamically during runtime.

D. Steps 5/6: Derivation of the Role Concept

By applying the first steps of the adapted SDRE model to the identified *MinaBASE*-using enterprise units, a hierarchy corresponding to a preliminary model of the role concept may be derived on the basis of the authorizations. The highest point is the administration that is not only responsible for administrating users and their assignment to roles, but also has all other authorizations in the system. The lowest point is the external guest, who is given fewest access rights. In between, the graph may be structured horizontally and vertically. Vertical structurization results from the

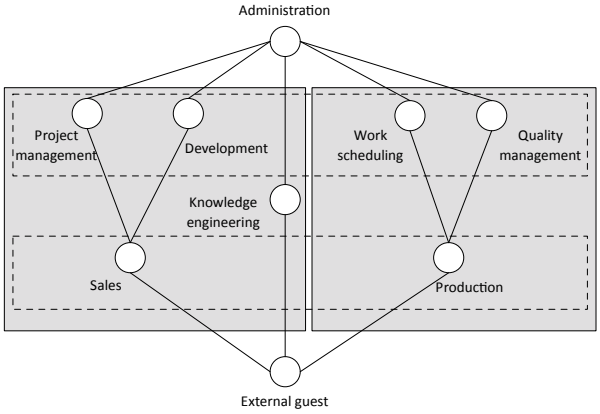

Figure 5. Preliminary role concept based on the permission catalog

degree of orientation to orders. This means that planning of working steps of a process chain and their execution are much more related to orders than the development of solution alternatives for a certain customer specification by technical experts. Horizontal structurization results from the authorization steps.

Then, the last step of the SDRE process follows, i.e., the analysis of the preliminary model for groupings of authorizations in the form of roles that exist several times and have a comparable amount of authorizations. These roles have to be eliminated. Otherwise, the catalog would list more roles than necessary, which might result in anomalies and undesired side effects in the administration of rights and roles. Review of the preliminary model taking into account the criteria described yields the role concept shown in Figure 6. Documentation of the authorizations for the individual company units shows that a separation between project management and development is not reasonable, as the access rights for the modeled scenarios and steps are identical. For this reason, both units are summarized by the developer role. The same applies to fabrication planning and

 ISBN: 978-1-61208-010-9

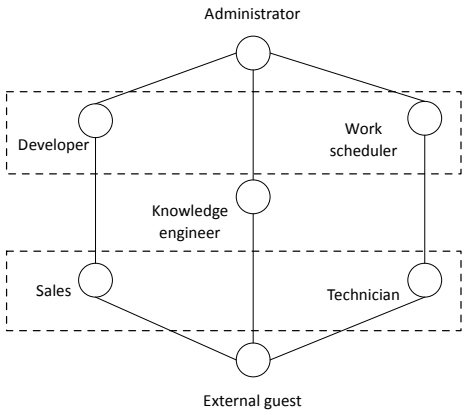

Figure 6. Revised role concept as RBAC model

quality management, as both units use *MinaBASE* for various objectives, but still have comparable use scenarios and, hence, identical authorizations. Consequently, they assume the same role in the use of *MinaBASE* , the role of the work planner. Below, the role of technician exists, who is responsible for the implementation of the plans made by the work planner. The technician is in the position to acquire the measurement data for the tests of the work planner and to define detailed parameters, such as machine instructions, as information added to process elements. To fulfill his task, the developer needs deep insight into the details of the competences and process chains, as he has to extend the strategic assessment of the sales division by a guaranteed technically possible feasibility assessment. The "knowledge engineering" component has already encapsulated the rights to update order-independent knowledge in the preliminary model. In this way, additional authorizations can be assigned specifically to a role.

V. Implementation in an IoC-Framework

This section describes the implementation of the RBAC model within *MinaBASE*. At first, the used framework, Spring Security, is introduced. After that, it is shown how static and dynamic security requirements stemming from the permission catalog as well as the constraint catalog can be fulfilled.

A. Spring Security

Spring Security is a subproject of the application framework "Spring" to control authentication and authorization in the JEE environment, i. e., in the range of business applications based on Java technology [13]. It is empowered by technologies provided by the core of Spring, such as "Inversion of Control" (IoC) using "Dependency Injection", which means a passive provisioning of an application component's dependencies by a central container known as the Spring "ApplicationContext", as well as the aspect-oriented programming (AOP) capabilities provided by that container. AOP is used for central encapsulation of cross-cutting concerns into so-called aspects, which avoids the scattering of duplicated code for realizing them across the codebase. By integrating with the hosting web container, a central hook is implemented by which a chain of filters can monitor and control the processing of HTTP requests as well as the execution of application components. This enables Spring Security to capture all elements of an application's architecture while thoroughly ensuring its security requirements using a declaratively configurable mechanism. This central hook determines whether an incoming request is trying to access a protected resource according to the supplied configuration. If this is the case, the "AuthenticationManager" (AM) is requested to authenticate and return the current principal, an abstract notion for, e. g., the currenlty logged in user, which is used by the "AccessDecisionManager" (ADM) to determine whether its role has the permission required for the protected resource in question. These two components can be controlled in a very flexible manner. For instance, during authentication, the AM can be configured to consult different providers, which in turn compare the principal's credentials by querying relational databases or LDAP directories. The ADM can be controlled by assigning static key/value-pairs of resources and required permissions or by enabling the dynamic execution of AOP-driven components. The following shows how Spring Security can be used to enforce compliance with the static and dynamic security requirements as specified in the permission- and constraint catalog in Figure 4.

B. Static aspects of security

Extending the security mechanisms across the entire architecture of *MinaBASE* requires the ADM to use a FilterSecurityInterceptor" (FSI) for securing the presentation layer as well as a MethodSecurityInterceptor" (MSI) for the application layer. To protect the application layer, a configuration of the MSI is required that determines which permissions the role of the current principal must possess to invoke components for data access and application logic. This can be realized by placing annotations directly in the application source code or through a central AOP configuration. The latter variant is used due to easier maintenance and therefore shown in Listing 1. For the protection of method invocations, a so-called pointcut, which is an entry point for the execution of code formulated as AOP-advices, is associated with a permission, whose presence will be checked by the MSI.

```
<global-method-security>
 <protect-pointcut
  expression="execution(*
   edu.kit.minabase.*CompetenceDAO.get*(..))"
  access="PERM_R_Competence"/>
</global-method-security>
```

Listing 1. Configuration of the MethodSecurityInterceptor

This restricts the data access to competences by requiring the presence of the ”PERM_R_Competence“ permission for the execution of methods, whose names start with ”get“ and are located within the CompetenceDAO class. The assignments of permissions to roles can be altered using an administrative interface at runtime. The invocation of methods for modifying and deleting information entities within *MinaBASE* as well as the execution of application logic can be restricted in a similar fashion. For securing the presentation layer, combinations of URL-patterns for protected regions and required permissions are specified, which are evaluated by the FSI during the monitoring of HTTP request processing. An excerpt of the necessary configuration is shown in Listing 2. Due to the URL-patterns being evaluated from top to bottom, the monitoring is at first disabled for static resources to achieve higher performance. Thereafter, permissions for visiting URLs matching the location for display and editing of competences are stated.

```
<http auto-config="false"
 access-denied-page="/denied.jsp">
 <intercept-url
  pattern="/static/*.*" filters="none"/>
 <intercept-url
  pattern="/competence/show/**"
  access="PERM_R_Competence"/>
 <intercept-url
  pattern="/competence/edit/**"
  access="PERM_W_Competence"/>
 <form-login login-page="/login.jsp"
  authentication-failure-url="/login.jsp?error=1"/>
 <logout logout-success-url="/logout.jsp"/>
 </http>
```

Listing 2. Configuration of the FilterSecurityInterceptor

The final step is the declaration of URLs, to which the AM will redirect unauthenticated users, that try to access a protected resource as well as URLs for authentication failures and the termination of a user's session. These settings ensure that protected areas are not reachable for users that dont possess the required permissions. To improve the user experience, links to sections the user does not have access to should not be displayed in the first place. To achieve this, the generation of HTML needs to be controlled with permissions in mind. Spring Security is bundled with an extension that allows fragments of Java ServerPages (JSP) to be rendered according to the current user's permissions, which is demonstrated in the following Listing 3.

```
<sec:authorize ifAllGranted="PERM_W_Competence">
 <a href="/cometence/edit/...">
   Edit this competence</a>
</sec:authorize>
```

Listing 3. Permission based generation of the user interface

This JSP-Tag assures that links to the area for editing competences are only rendered to those users that have the ”PERM_W_Competence“ permission. These three listings show how Spring Security can be used to employ a

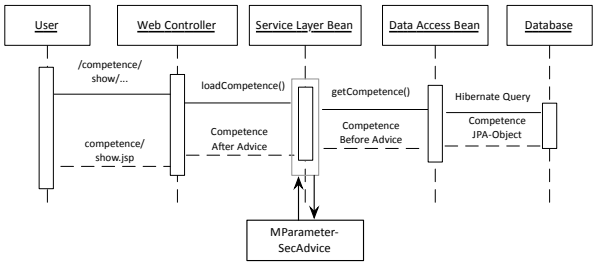

Figure 7. Integration of the MParameterSecAdvice in *MinaBASE*

homogeneous system of permissions that stems from the SDRE permission catalog and covers the entire application architecture from data access to the presentation layer. At runtime these permissions are assigned to roles from the designed RBAC model.

C. Dynamic security aspects

In the previous section, security aspects were considered, which could be fulfilled by statically restricting access to a protected resource by requiring a specific permission to be held by the current principal. While most aspects of the permission catalog are covered by this approach, entries of the constraint-catalog as depicted in Figure 4 cannot be implemented in this fashion, because of their dynamic nature, wich means, these constraints cannot be enforced at build-time, but only at runtime. As an example, the filtering of Fabrication-specific TP of a competence's detailed view is used. To avoid code duplication whenever Fabrication-specific TP of a *MinaBASE* information entity shall be filtered, this concern is encapsulated into a separate AOP-Advice called ”MParameterSecAdvice“, whose integration into the method's call flow is illustrated in Figure 7. A request for the detailed view of a competence is received by a Web-controller, which initiates the data access for the current competence by invoking methods from the service layer. Once this competence is loaded as a database object, the MParameterSecAdvice is hooked into the execution flow using AOP-Weaving. The job of this component is to iterate over the competence's parameter collection and filter out those parameteres to which the current principal has no permission. The revised competence object is then returned to the controller which starts the generation of HTML templates and sends the result to the browser. The advantage of this appraoch is the fact that the permission based filtering is encapsulated into the MParameterSecAdvice once and can be applied declaratively to multiple application components without code duplication and mixture of concerns by simple configuration in a similar fashion as described in Listing 1.

 ISBN: 978-1-61208-010-9

VI. Conclusion

In this paper, a role concept for the process knowledge database *MinaBASE* has been developed based on a systematic methodology called Scenario-driven role engineering. The implementation of this role concept within an IoC-Framework such as Spring has been demonstrated by utilizing Spring Security and technolgies such as AOP. At first, the *MinaBASE* approach as well as the Scenario-driven role engineering methodology were introduced. Following this, careful adjustments were made to the inputs of the SDRE process resulting from the background and purpose of *MinaBASE* without hurting the methodology's idea and principles. Then the application of the SDRE process was shown including examples on how to derive a minimal set of permissions enabling each role to fulfill its work profile. In the following section the implementation of the derived role concept using Spring Security is described in detail. Important concepts are Dependency Injection and AOP, as they enable Spring Security to ensure static and dynamic security requirements across the entire application architecture. For the implementation of security requirements that can be decided at runtime only, an example was given in order to prevent the disclosure of Fabrication-specific parameters for non-authorized persons.

References

[1] U. Hansen, C. Germer, S. Büttgenbach, and H. Franke, "Rule based validation of processing sequences," in *Techn. Proc. MSM*, 2002.

[2] D. F. Ferraiolo and R. Kuhn, "Role-based access control," in *In Proceedings of 15th NIST-NCSC National Computer Security Conference*, October 1992, pp. 554–563.

[3] G. Neumann and M. Strembeck, "A scenario-driven role engineering process for functional RBAC roles," in *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2002, pp. 33–42.

[4] I. Nonaka and H. Takeuchi, "The knowledge-creating company," *Harvard Business Review*, vol. 6, pp. 96–104, 1991.

[5] M. Dickerhof, "Prozesswissensmanagement für die Mikrosystemtechnik." *Statusseminar MikroWebFab, Karlsruhe*, 2003.

[6] M. Dickerhof and A. Parusel, "Bridging the Gap—from Process Related Documentation to an Integrated Process and Application Knowledge Management in Micro Systems Technology," *Micro-Assembly Technologies and Applications*, pp. 109–119, 2010.

[7] M. Dickerhof, O. Kusche, D. Kimmig, and A. Schmidt, "An ontology-based approach to supporting development and production of microsystems," *Proc. of the 4th Internat. Conf. on Web Information Systems and Technologies*, 2008.

[8] D. Kimmig, A. Schmidt, K. Bittner, and M. Dickerhof, "Modeling of microsystems production processes for the minabase process knowledge database using semantic technologies," *Proc. of the 3th Internat. Conf. on Information, Process, and Knowledge Management*, 2011.

[9] E. J. Coyne, "Role engineering," in *RBAC '95: Proceedings of the first ACM Workshop on Role-based access control*. New York, NY, USA: ACM Press, 1996, p. 4.

[10] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," in *Inproceedings of fourth ACM Symposium on Operating System Principles*, 1975.

[11] HL7 Security Technical Committee , "HL7 Role Based Access Control (RBAC) Role Engineering Process," January 2005.

[12] K. Boehm, W. Engelbach, J. Härtwig, M. Wilcken, and M. Delp, "Modelling and implementing pre-built information spaces. architecture and methods for process oriented knowledge management," 2005.

[13] B. Alex and L. Taylor, "Spring Security Reference Documentation," http://static.springsource.org/spring-security/site/docs/3.1.x/reference/springsecurity-single.html, last access: 06.04.2011.

 ISBN: 978-1-61208-010-9