Efficient Access Control for Composite Applications

M. Wimmer[1], M.-C. Albutiu[1], A. Kemper[1], M. Rits[2], and V. Lotz[2]

[1] Technische Universität München, 85748 Garching b. München, Germany

[2] SAP Research, Font de l'Orme, 06250 Mougins, France

{wimmerma, albutiu, kemper}@in.tum.de, {maarten.rits, volkmar.lotz}@sap.com

1 Motivation

Composite applications rely on further sub-applications – also called sub-activities in the following – to implement their functionality. There are numerous examples including quite simple Web applications as well as large scale enterprise resource planning (ERP) systems that interact with database backends. Also, business processes that are realized as Web service workflows represent complex composite applications. Thereby, sub-activities can constitute composite applications themselves.

In general, sub-applications are self-contained software modules that autonomously enforce their own security policies. This autonomy of authorization can lead to significant performance drawbacks: On the one hand, the authorizations of legitimate users are evaluated repeatedly. On the other hand, requests of ultimately unauthorized users that lack authorizations at later stages of the workflow can lead to transaction rollbacks or demand for compensating transactions. Thus, it appears beneficial to evaluate the authorizations of users as soon as possible by shifting access control to the workflow layer instead of retaining it at the sub-activities. Regarding composite applications this can be a non-trivial task, as the access control configurations of several autonomous sub-applications have to be taken into account. The key to success is a consolidated view onto the access control of composite applications providing answers to the following questions: (1) What are the least required privileges?, (2) Who is allowed to execute the composite application?, and (3) Are there possibilities to reduce policy evaluation costs?

The first issue addresses the principle of least privilege, denoting that only those privileges are granted which are required in the context of the sub-activities. Following this design paradigm reduces security vulnerabilities as it guarantees that no business resources other than the ones needed by the composite application can be accessed. As we showed in [WEK05, WEFK05], this restriction is of particular importance for the design of Web services that interact with database systems.

Knowing the group of authorized users allows to detect unintended configurations more easily. For instance, if only highly privileged users like managers are authorized to execute a business process, this might be an indication that the composite application itself has to be revised. We are addressing this issue from the *single-user / single-role* perspective, meaning that a user can execute the application by

the activation of one task specific role. This complies with many business processes which are typically representing job specific tasks. Therefore, composite applications are to be distinguished from multi-user workflows which are business processes that are executed by several users in a team.

Optimization capabilities for composite applications – as addressed by the third issue – can be given in two ways: On the one hand, a consolidated policy allows the early-filtering of requests. Application invocations which will lead to aborts at later stages in the process due to missing privileges can be detected and averted. On the other hand, repeated and redundant authorization checks by the individual sub-activities can be omitted, in case the authorization decision can be inferred on the composite application's layer.

In this contribution, we show how consolidated policies of single-user workflows can be generated. This optimization technique has been integrated into SAP Research's workflow management tool suite which allows to compare traditional and optimized policy evaluation strategies.

2   Consolidating the Access Control of Composite Applications

In order to consolidate the access control of composite applications, the workflow structure, dataflow dependencies, and external dependencies have to be taken into account. Details about the consolidation process have been described in [WAK06] and [WKRL06]. The workflow structure defines the control flow, i.e., the execution order of the sub-activities as illustrated in Figure 1(a). From an access control point of view, sequential or parallel executions denote that all sub-activities are invoked. We represent this characteristic through the *sequence* pattern. Furthermore, conditional and event based executions are possible which – from the access control perspective – denote that only one sub-activity will be invoked. We represent this aspect through the *switch* template. The access control dependencies of a composite application can then be represented by means of a tree as illustrated in Figure 1(b). The composite application's policy is generated through a bottom-up analysis, combining the policies of the individual sub-activities. Users need to be granted execution privileges by all policies that apply to the sub-activities in order to be able to execute a *sequence* pattern. That is, the combined policy for a *sequence* pattern consists of the intersection of subjects and the union of all privileges defined in the policies of the autonomous tasks.

Regarding *switch* patterns, two different approaches can be applied: Following the *full-authorization* approach, users have to be authorized to perform all sub-activities, irrespective which one would actually be executed. Hence, policies are combined in the same way as for *sequence* pattern. In contrast to this, the *partial-authorization* approach considers each execution path individually. Regarding the *switch* node in Figure 1(b), users can execute the left branch in case they are authorized by policy $P_{3,4}$ and they can execute the right branch if they are authorized by P_2. Consequently, separate policies will be generated for the different execution paths of a workflow.

The resulting consolidated policy includes those privileges needed to execute the

(a) Control flow of a composite application   (b) Workflow tree representation
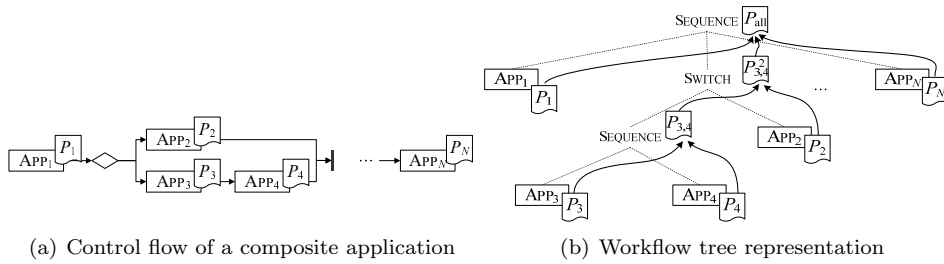
Figure 1: Consolidated policies are calculated by a bottom-up analysis

composite application, respectively workflow path. Thus, in case the policies of the sub-applications realize the principle of least privilege, this paradigm can also be inferred for the combined policy. Furthermore, the consolidated policy constitutes the basis for an optimized access control. If the full-authorization approach is applied, authorization checks can be shifted to the workflow management system (WFMS) and be omitted at lower execution levels. Thus, policy evaluation costs can be saved significantly. In case of the partial-authorization approach, authorizations for the individual workflow branches have to be evaluated separately, meaning that the WFMS has to be capable of enforcing access control at workflow branches. Both approaches help to avoid situations that demand for transaction rollbacks and compensating transactions. This is because requests are filtered before the execution of a composite application (full-authorization) or before entering a workflow branch (partial-authorization), so that ultimately unauthorized requests are detected as soon as possible.

3 Demonstration

The policy consolidation approach has been integrated into SAP Research's workflow management tool suite, including the three components Maestro, Nehemiah and Gabriel. Maestro is used to model business processes, by defining a set of sub-activities and their interdependencies, i.e., the control flow. Via drag-and-drop, components like sub-activity nodes or control flow nodes can be inserted and connected. Nehemiah is the workflow management engine that allows to execute business processes which have been designed using Maestro. At runtime, Nehemiah allows to supervise the state of the workflow by keeping track of active sub-activities. Thus, Maestro and Nehemiah are used for workflow modeling and activation. Individual sub-activities, on the other hand, are modeled and activated by Gabriel. Gabriel allows to specify the roles needed to execute sub-activities. At design time, sub-activity profiles are defined that describe which actions have to be performed when executing a certain task. For instance, an action can be the invocation of a Web service. When modeling a workflow with Maestro, sub-activity nodes can be associated with the corresponding sub-activity by means of the profile. Furthermore, subjects like roles and users can be modeled and these subjects can be granted the privileges required to execute respective sub-activities. The relationships between the three programs are illustrated in Figure 2.

Nehemiah supports the execution of multi-user workflows, denoting that the sub-
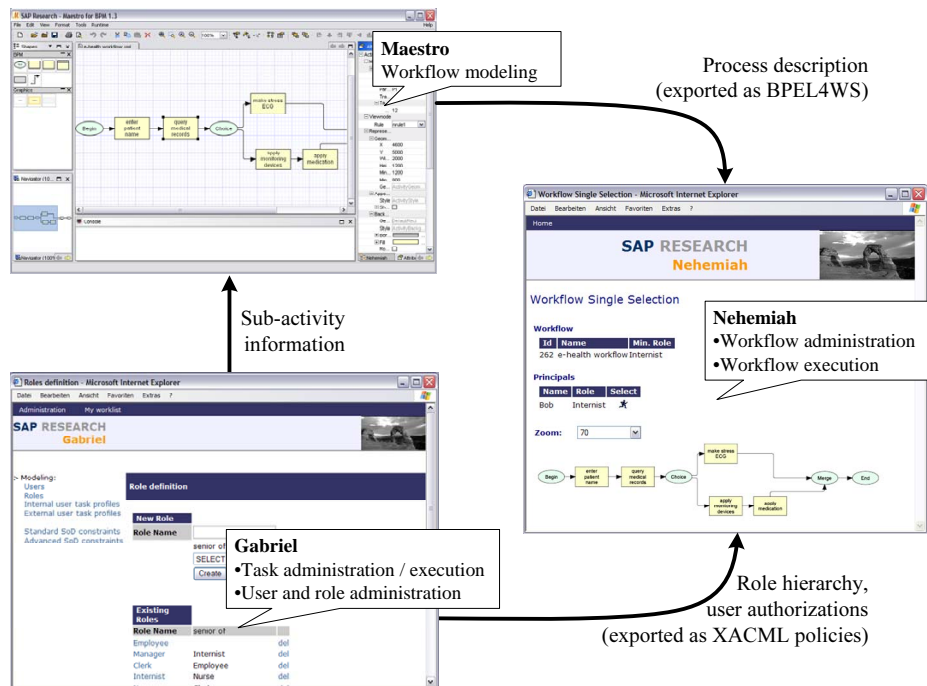
Figure 2: Integration into SAP Research's workflow management tool suite

activities can be executed by teams. We complemented Nehemiah's policy enforcement strategy with the special treatment of single-user workflows and composite applications. For this purpose, we integrated the full-authorization approach. In the course of the demonstration, the theoretical backgrounds of our policy consolidation approach are presented and illustrated by means of use cases. The consolidation of Web service policies (coded in form of XACML policies) is demonstrated and the optimized single-user execution is compared to the traditional approach that relies on separate policy enforcements.

References

[WAK06] M. Wimmer, M.-C. Albutiu, and A. Kemper. Optimized Workflow Authorization in Service Oriented Architectures. In *Proceedings of ETRICS '06*, volume 3995 of *LNCS*, pages 30–44, Freiburg, Germany, June 2006.

[WEFK05] M. Wimmer, P. Ehrnlechner, A. Fischer, and A. Kemper. Flexible Autorisierung in Datenbank-basierten Web Service-Föderationen. *IFE*, 20(3):167–181, December 2005.

[WEK05] M. Wimmer, P. Ehrnlechner, and A. Kemper. Flexible Autorisierung in Web Service-Föderationen. In *Proceedings of BTW '05*, pages 185–204, Karlsruhe, Germany, February 2005.

[WKRL06] M. Wimmer, A. Kemper, M. Rits, and V. Lotz. Consolidating the Access Control of Composite Applications and Workflows. In *Proceedings of DBSec '06*, volume 4127 of *LNCS*, pages 44–59, Sophia Antipolis, France, August 2006.