Document and Workflow Management in Collaborative Systems

Marius Ioan PODEAN
marius.podean@econ.ubbcluj.ro
Babeş-Bolyai University of Cluj-Napoca

Collaborative systems process a great amount of documents witch imposes restrictions regarding querying and integration. Using XML technologies facilitates this aspect and provides the means to easily integrate with standards that support workflow management. The model presented by this paper tries to integrate XML technologies with concepts like role based access control and workflow management in order to obtain an efficient user and document centric collaborative system concerned with the management of documents.

Keywords: *document management, workflow, RBAC, XML.*

In the context of collaborative applications an important aspect is data availability and efficient processing capabilities and the existence of the proper means to combine data. Data is often stored in databases and in different documents, making access difficult and reducing search capabilities by relevance degree. Another aspect is data availability in different formats which do not allow automatic mean determination by computers (HTML) and do not support reasoning. The use of microformats has underlined the importance of open standards and especially the great efficiency obtained in data processing when combining data with semantics that are commonly used.

In collaborative systems a great amount of data is manipulated using documents, which are most often stored in different proprietary formats. This aspect makes impossible multiple documents querying in order to obtain relevant information and generating new documents. Document access policies impose different restrictions in organizations with a large number of members where they can easily change positions in the organization. An important issue is to ensure the means to access documents from outside the organization by computers not just human users, which requires very flexible access policies. Documents flow in the organization can generate great time loses if the operations are not automated and can imposes restrictions upon users regarding software and OS if proprietary formats are used.

This paper presents a model that tries to integrate these issues and bring and possible solution regarding document management and there workflow in collaborative systems, taking in consideration the semantic web which emphasizes the addition of meaning to data and ensuring data integration and interoperability. This model addresses requirements concerning access control using the Role-Based Access Control (RBAC) model, document workflow and the physical level where documents are stored based on XML technologies. This paper represents a work in progress addressing mainly design aspects and suggesting means of implementation. The first section of the paper will present more detailed the main architecture of the model, following that in section 2 the access control model to be addressed. In section 3 the document workflow subcomponent is broth in to attention and in section 4 document management and document storing elements are emphasized. The final section underlines a series of challenges identified that can influence the implementation of the model.

1. General architecture

The architecture of the model is mainly based on the concept of loose coupling which allows binding its components without generating dependencies between them, all components having a communication layer to ensure message exchange. This concept allows changes inside the component without influencing or imposing restrictions upon components interaction.

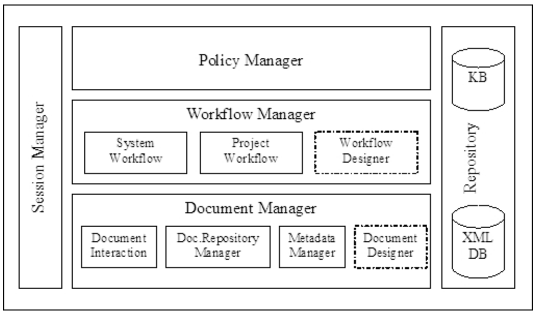

Fig.1. General architecture

The *Policy Manager* component represents the layer that ensures system and document access policies management implementing a model based on roles. This model works with concepts like roles, permissions, work sessions, objects and operations and also static/dynamic separation of duty and hierarchic levels covering most of the aspects regarding the internal structure or a collaborative system.

The *Workflow Manager* component addresses document workflow inside the system and certain aspects regarding the automation of document access and manipulation. Besides defining the order in which operations can be executed upon a set of documents that belong to a particular project in order to process them, aspects concerning the access of different objects from the system taking in consideration the role assigned by the access policy manager when initiating a work session. This aspect allows system configuration in order to meet a vast variety of needs without altering component independence and there interoperability.

The *Repository* component represents the physical layer of the model handling the storage of XML documents, their definition schema (XSD), transformations models (XSL), queries that can be applied to documents (XQuery), user interaction elements (XForms) and generation graphical interfaces (CSS). This level implements also the knowledge base required by the policy manager to ensure user authorization.

The functionality of the whole model is ensured by the *Session manager* component witch implements the interconnection and the interoperability of the aforementioned components. At this level the user authentication occurs and by consulting the policy manager component the authorization takes place and a role and a set of permitted operations are assigned to the current work session. Based on identity and the assigned role, by consulting the workflow manager it will be determined the set of actions that can be executed inside the system and especially in the projects to witch the current user is a member. The set of actions is determined taking in consideration also the state of evolution of the project. Based on the assigned role and the set of permitted actions using the document interaction interface the current user can execute the desired actions upon documents. The session manager must provide the proper means to be able to satisfy requests triggered by human users and also by computers, this layer representing the communication interface with the outside environment.

2. Policy Manager

The access policy manager represents the level that decides upon the set of permitted operations that can be executed by a particular user during a certain work session. This level implements the RBAC model, witch is based on concepts like users, roles, permissions, objects and sessions. Users are assigned to a particular role and roles have a certain set of permissions associated, users accessing permissions in virtue to the assigned role. Users interact with the system during a work session and roles are assigned to these sessions. Roles represent m-n relationships between users and permissions. The permissions represent the authorization to execute a set of actions upon protected object. A great advantage of this model is that it allows the definition of hierarchic levels inside the system, being able to emphasize the authority and responsibility layers. This aspect imposes certain restrictions concerning permissions multiple inheritance witch can lead to interest conflicts (a certain user should not be allowed to authorize a personal expenditure). In order to eliminate these issues, the RBAC model introduces the concept of static/dynamic separation of duty.

Static separation of duty requires the definition of certain restrictions in order to impose that a user can activate just one role during a particular work session, while dynamic separation of duty accepts the fact that users can activate multiple role during a work session but imposes the restriction that the activated roles should not be in conflict in the means of the allowed permissions. Using both static and dynamic separation of duty allows the definition of highly flexible access policies that can realistically simulate real life organizational model.

Implementing such a model requires certain means to express relations between elements and reasoning capabilities based on these relationships. One possible manner of implementation is based on OWL language witch is specialized on defining ontology along with the DL subset to define descriptive logic rules. This implementation aloes the definition of highly dynamical access rules and provides the proper means to easily represent dynamic separation of duty even in the context of hierarchic organizational structuring. By using languages like SPARQL queries can be executed based on ontology and the according knowledge base, but the level of complexity imposed by the definition of correct and complete ontology combined with a particular lack of tools required to integrate with the rest of the system imposes a set of restrictions regarding implementation. The RBAC model can be also implemented using XML Schema combined with Schematron, witch is a specialized language used to define validation rules for XML documents, allowing to create assertions based on available data. The fact that this approach is supported by a great set of tools makes it more favorable to be used in our implementation, but it also imposes certain design and extendibility restrictions witch are not compensated by the ease of use.

3. Workflow manager

Workflow management is modeled on two levels, taking in consideration the administrative task required to ensure the maintenance of the system and the main level that covers the flow of documents inside the projects. Document workflow management handles the operations that can be executed on sets of documents, their required order necessary to obtain new documents or to pass the existing ones from one state to another. The structuring unit is the project, witch can consist in one or more documents upon witch a set of actions can be executed by one or more users (including here not just human users but also computers that access the system), operations that do or do not follow a particular execution order.

Conceptually the flow of documents can be described very easily by using finite state machines or Petri nets. The document workflow must describe the order that operations must follow when a particular user is working on a set of documents at a certain moment of time taking in consideration the state in witch the set of documents is at the moment. This requires for a project the existence of a model that describes the general workflow associated with the roles required and also the restriction imposed to different roles (if the project workflow requires them) according to project states. To synthesize, the project workflow management subcomponent must handle document transitions and the implied roles.

The model of this component can be implemented using standards like WS-BPEL, WfML or YAWL. From these alternatives Web Services Business Process Execution Language presents a set of advantages over the other languages, like that that it is largely used and it is backed-up by large IT companies witch in a collaborative context where the availability and the need to properly understand workflow elements even by external partners is an important element makes it a proper implementation. Functionally, WS-BPEL allows the workflow implementation disregarding the manner in witch these have been conceptually described (finite state machines or Petri nets), providing even a personal manner for abstract process description (abstract business processes). The fact that the implementation of this standard represents an orchestration language centered on the workflow execution makes it the most attractive implementation alternative even thou the communication is based on web services interfaces.

4. Document Manager

This component represents the level where aspects like document retrieval, attachment of the metadata layer or interaction with other components is implemented.

The *Document Interaction* subcomponent assures the interface to the implementation of the according operations that can be executed in order process documents, this mean representing the only manner that documents can be accessed from outside the component. This approach come to sustain the concept of loose coupling used in the system and allows even computer interaction with documents (a.i. querying public documents). This level implements all operations defined by the access policy manager.

Taking in consideration that all documents are stored in XML format, it is imposed the implementation of a subcomponent to handle all documents implied by a project. This so called helper documents represent document definitions (XSD), transformation export to other formats (XSL, XSLT-FO), definitions of allowed document queries (XQuery) and document interaction interfaces (XForms). These documents are stored in a XML native database, the *Document Repository Manager* subcomponent providing the proper means for retrieving and querying them inside a project.

In the semantic web context, providing content that can easily be understood by both computers and browsers is a must. In order to model this requirement information and documents that are accessible from outside the system have an additional layer of metadata provided by using microformats RDF based document descriptions. The *Metadata Manager* subcomponent attaches different semantics (hCard, hReview, hResume, FOAF) and automatically generates the required RDF based document descriptions.

5. Conclusions and future work

In order to provide a very powerful and flexible system, requirements like testing the available technologies required for implementation and deciding on the most efficient ones is crucial. The system must also be very flexible in respect to user and organizational environment requirements, providing the necessarily means to extend the system. To ensure these requirements, two extension points are proposed: Document Designer (must allow the definition of new document types and necessarily operations) and Workflow Designer (defines the workflow of the new created document types). These components are proposed as extension points because their need is conditioned by the efficiency of the aforementioned model.

Acknowledgement

This work was partially supported by National Program 2 Grants no.92-100/2008. (SICOMAP-Collaborative systems for the management of economic projects)

References

• Ayed, S., Cuppens-Boulahia, N., Cuppens, F. (2008), Deploying Access Control in Distributed Workflow. ACSC2008, Wollongong, NSW, Australia.

• Bertino, E., Ferrari, E., Alturi V. (1999), The Specifications and Enforcement of Authorization Constraints in Workflow Management Systems. ACM Transactions on Information and System Security.

• Ferraiolo, D., F., Sandhu, R., Gavrila, S. (2001), Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, Vol. 4, No. 3.

• OASIS (2007), Web Services Business Process Execution Language Version 2.0, OASIS Standard 11 April 2007

• Workflow Management Coalition (1995), The Workflow Reference Model