INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL.11, NO. 4, DECEMBER 2006，272-283

An RBAC Model-Based Approach to Specify the Access Policies of Web-Based Emergency Information Systems

Ignacio Aedo, Paloma Díaz and Daniel Sanz

Abstract- **One of the main design challenges of any Web-based Emergency Management Information System (WEMIS) is the diversity of users and responsibilities to be considered. Modelling the access capabilities of different communities of users is a most relevant concern for which the RBAC (Role-Based Access Control) paradigm provides flexible and powerful constructs. In this paper we describe how we used an RBAC model-based approach to specify at different levels of abstraction the access policy of a specific WEMIS called ARCE (“*Aplicación en Red para Casos de Emergencia*”). This approach made possible to face access modelling at earlier development stages, so that stakeholders got involved in analytical and empirical evaluations to test the correctness and completeness of the access policy. Moreover, since the RBAC meta-model is embedded into a web engineering method, we put in practice a holistic process addressing different design perspectives in an integrated way.**

***Index Terms*— Emergency management system; role based access control; web engineering; user-centred design.**

1. INTRODUCTION

One of the main design challenges in a Web-based Emergency Management Information System (WEMIS henceforth) is the great variety of users and responsibilities that have to be considered. One the one hand, there are different kinds of virtual communities involved in the emergency situation with different needs and natures: from the teams set up at the governmental agencies -where technical specialists, experts in different related areas and strategic and political representatives cooperate following well defined protocols and rules-, to less structured communities of users like invited organizations, ngo’s, journalists, volunteers, anonymous users, etc. On the other hand, it is commonly recognized that each user can exercise one or more responsibilities during the emergency situation [1, 2] so that the system has to support flexibility whilst guaranteeing security and auditing. In such a scenario, modelling the access rights of different users communities in a proper way becomes a most decisive design concern. For this purpose the Role Based Access Control (RBAC) paradigm [3] is a most convenient option to establish access policies in a flexible and easy to maintain way [4, 5]. The key RBAC hypothesis is that roles (i.e. responsibilities) are much more persistent than users. Once the responsibilities of an organization are defined they hardly change, what usually changes is the user or users that exercise a specific responsibility in a specific situation. Consequently, an access policy based on assigning permissions to the roles and then allocating users into the appropriate roles is much more stable and easy to maintain than user or group-based approaches like DAC (Discretionary Access Control) or MAC (Mandatory Access Control). In this paper we will focus on how to use the RBAC paradigm in the context of WEMIS to specify in a progressive and integrated way the access rules. We will not deal with cooperation issues such as how to define the adequate roles to achieve effective cooperation as in [6] taking advantage of the core premises underlying the RBAC paradigm so that we can offer the basic components to establish adequate access policies. We understand efficiency as the easiness to define and maintain access rules as well as the easiness to check such rules with stakeholders.

Another important issue to keep in mind is that access modelling has to be integrated within the rest of the modelling activities to produce a well-documented and safe system. In this way, access requirements are unified with other kinds of requirements (such as functional and non-functional requirements) and are easy to test, maintain and re-use [7]. When such a *holistic* engineering approach is not adopted, access requirements are often added to the system once it has been implemented, a process that is costly and error-prone, since access policies have to be shoehorned into existing code [8]. For instance, there are some UML extensions aimed at unifying security with other requirements including [9, 10, 8]. In the particular case of web systems the unique development method integrating access requirements is the Ariadne Development Method -ADM- [11, 7]. In this work we describe how we applied the model-based approach of the ADM method to deal with the specification of the access needs of different virtual communities of users of a specific WEMIS called ARCE (“*Aplicación en Red para Casos de Emergencia*”) at different levels of abstraction and following a *holistic* perspective. At the highest level of abstraction rules are expressed in a way that is closer to the stakeholders’ perspective, so that they can be assessed with them before any code is produced. At the lowest level of abstraction rules are closer to implementation units so that consistency checks can be performed as well as automatic generation of code.

The remaining of the paper is organized as follows. First, we give a short introduction to the concept of RBAC and shortly discuss its main benefits. Then we describe the model-based approach to specify access rules at different levels of abstraction. Section 4 is devoted to the description of the ARCE system and section 5 focuses on how the

This work has been funded by projects ARCE ++ (TSI2004-03394) and MODUWEB (TIN2006-09768) supported both by the Spanish Ministry of Education and Science. ARCE is being developed in collaboration with “Dirección General de Protección Civil y Emergencias” (Spain). This paper is an extended version of “Modeling Emergency Response Communities using RBAC Principles” published at *ISCRAM 2006 International Conference, Newark, NJ*, USA, May, 2006.

I. Aedo, Díaz, P and Sanz, D. are with DEI Laboratory at Carlos III University of Madrid, Ave. de la Universidad, 30. 28911 Leganés, Spain. email: {aedo@ia|pdp@inf|dsanz@inf}.uc3m.es

model-based paradigm has been applied to specify and assess the different access requirements of this system. We finish by drawing some conclusions in the form of design guidelines we have derived both from the literature as well as from our experience in developing this system applying a web engineering method.

2. RBAC IN A NUTSHELL

RBAC regulates the access to resources based on organizational entities called roles [3]. The key idea is that, instead of granting privileges directly to individual users, privileges are assigned to roles, and users are made members of adequate roles. A role represents a job function or a set of responsibilities for a set of users holding that role, together with the privileges granted to it. Privileges are granted to roles by adding permissions for that role using the permission-assignment relationship (*Has* relationship in figure 1), while users are made members of roles by means of the user-assignment relationship (*Assumes* relationship in figure 1). Permissions can be specified at two levels of abstraction. High-level permissions are composed by a function (e.g. *read a new*), as in [5] so that access rules are close to the stakeholders' view of the system (e.g. <*L9, read a new*>). Low level rules are defined in a more general way [3] as a system object (e.g. a *file*), and an operation defined for that object (e.g. *read*), leading to access rules such as <*L9, read, file=new07.xml*>. In this case, objects are the resources to be accessed by users, and operations represent all actions that can be performed on the system. It is important to note that the nature of permissions is left open in RBAC, which implies that the security designer needs to specify object and operation granularity as part of the system design.

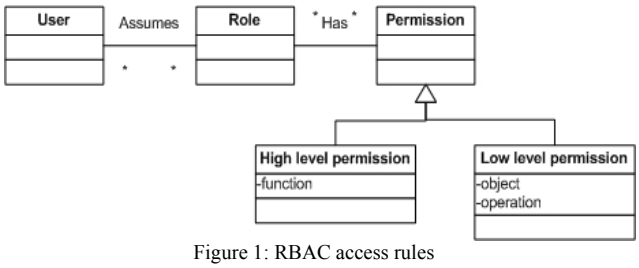

Figure 1: RBAC access rules

When a user enters the system (after authentication), a session is created, which includes the user and the set of active roles. This set contains one or more roles from the set of roles assigned to the user ("Assumes" relationship in figure 1), and represent all permissions available to the user for the session. RBAC extensions include role hierarchies and constraints. Role hierarchies allow senior roles (more powerful in terms of permissions) to inherit permissions assigned to junior roles (less powerful). With the addition of hierarchies, the role structure becomes a tree or a graph used to reflect organization's lines of authority, competence and responsibility in a natural way [4]. Constraints represent limitations imposed to some RBAC elements, normally the user-assignment relationship. The most common is the separation of duty (SoD), which refers to the partition of some critical tasks, splitting the permissions needed to perform them into several roles that are mutually exclusive, so that a single user cannot be assigned simultaneously to these roles. Thus, in order to accomplish a critical task, more than one user is required, ensuring that fraud can not occur without deliberate collusion of several users.

In the web arena, where the number of users becomes unmanageable, RBAC mechanisms are a most convenient approach to deal with security rules in an efficient and easy to maintain way [5]. Firstly, permissions are expressed using roles and operations that belong to the domain of application so that they are close to the organizational perspective and, therefore, it will be easier to test with stakeholders their correctness and completeness. And secondly, access rules are more stable since in any organization roles hardly ever change, what changes is the user or users that hold such a role. As rules depend on roles they tend to be quite persistent and the less changes we have to do to maintain our policy, the more robust and less prone to error it will be. Indeed some empirical studies demonstrate that RBAC policies make possible to reduce considerably the management efforts [12, 13].

In the particular case of WEMIS, where roles have to be clearly defined in terms of the access privileges they have and users have to be trained to assume different roles during the crisis [2], the RBAC paradigm seems to be quite adequate. Access rules defined as <role, operation> will be specified at design time so that the operation protocol can be clearly established and tested with the stakeholders. This policy will hardly change, only if new functions or responsibilities are identified and then a new tuple can be added. Each specific user is assigned all the roles she's allowed to play and at runtime the user just authenticates herself and accesses to all the allowed functions. RBAC policies do not impose any kind of constraint in the number or kind of roles a user can be assigned to, and even they can be assigned at runtime since this operation does not interfere with the core of access policy that depends on the roles. Thus, users could assume any role during the emergency situation as far as they are trained to exercise the responsibilities of such role using the WEMIS.

The expressiveness and flexibility of RBAC were the main reasons to apply this paradigm to cope with the access requirements of the different users communities of ARCE, as opposed to other policy-dependent access control mechanisms such as DAC or MAC. ARCE is a WEMIS supporting international cooperation in emergency situations as shown in section 5. The MARAH meta-model (see section 3.1) can be used to specify the rules that ensure a proper operation of any kind of web system assuming an RBAC philosophy in order to simplify management and avoid inconsistencies and errors. However, using a formal model to specify the access rules isn't a suitable option for many developers who will prefer a number of conceptual models closer to the stakeholders' perspective of the problem. Moreover, such design models should be integrated with the rest of the system

requirements to support a holistic development process. In this context, we propose the use of a model-based approach that relies on a formal meta-model. Thus while MARAH provides the formal components (e.g. roles, teams, objects, composition mechanisms, users, authorizations and so on) ADM offers a number of design models based on such components (e.g. users diagrams, structural diagrams, authorization rules, user allocation and son on) that make it possible to specify the policy at different levels of abstraction by means of a number of easy to understand diagrams (see section 3.2). Using the ADM, access modelling is addressed at the earliest development stages so that stakeholders can be involved in analytical and empirical evaluations to assess the access policy before the system is already implemented. Moreover it is possible to integrate access modelling with other design views addressed by the ADM, such as information structure, presentation features, navigation capabilities or interaction mechanisms. A complete description of MARAH can be found in [14, 7] and ADM is fully presented in [11].

3. A MODEL-BASED APPROACH FOR ACCESS MODELING

Model-based development is a software development paradigm that has been gaining in popularity in recent years due to the OMG group effort and, and particularly to their MDA (Model Driven Architecture) proposal [16]. This paradigm aims to move the development effort from implementation towards the creation of models based in some meta-model from which code can be generated automatically or semi-automatically. The fundamental value of formal meta-models resides in their capacity to guarantee platform independence from implementation and, therefore, for facilitating the interoperability among heterogeneous systems. This development paradigm turns out to be suitable in the case of the web systems in which interoperability and flexibility are inescapable requirements. In this section we describe a model-based approach for access modeling which consists of two levels of meta-models: at the highest level of abstraction there are some design meta-models used by developers to specify access needs; at the lowest level of abstraction a meta-meta-model called MARAH gathers all the components and functions in a unique formal expression. While the higher abstraction model offers an intuitive tool for access control modeling that can be used as a powerful communication mechanism with stakeholders in order to promote a participatory design; the lower abstraction model, that captures fine grained access requirements, allows to formally check policy properties and is closer to the system implementation, facilitating the automatic transformation to implementation units. This model-based approach will be used to model the access requirements of virtual communities of a WEMIS in a flexible and easy to understand way as it will be shown in section 5.

3.1 The MARAH Meta-Model

The MARAH [14] formal model can be used to specify the rules that ensure a proper operation of any hypermedia/web system. Hence, one of its basic assumptions is to use abstractions and concepts that belong to the hypermedia domain in a broad sense, so that the model can be integrated into a hypermedia design method. MARAH follows an RBAC philosophy in order to simplify management and avoid inconsistencies and errors.

From a structural point of view, the MARAH model is composed by a number of elements and relations among elements (see table 1) and that are shortly described below.

Table 1. The MARAH formal model

Abstract Element	Specification
Subject Roles Teams	$S= \{R \cup T \mid R \cap T = \varnothing\}$ $R=\{r_i , i=1..n, n\in \mathbf{N}\}$ $T=\{t_i , i=1..m, m\in \mathbf{N}\}$
Users	$U=\{u_i , i=1..p, p\in \mathbf{N}\}$
Users assignment	$A=\{<u,r> \mid u \in U, r \in R\}$
Separation of duties	$SD=\{<r_i, r_j> r_i, r_j \in R, r_i \neq r_j \mid$ $\forall u_k \in U, < u_k, r_i > \in A \Rightarrow$ $< u_k, r_i > \notin A\}$
Objects Nodes Contents	$O =\{N \cup C\} \mid N \cap C= \varnothing$ $N=\{n_i , i=1..q, q \in \mathbf{N}\}$ $C= \{c_i , i=1..r, r\in \mathbf{N}\}$
Access categories	$SC =\{sc_i , i=1..3 \mid sc_i \subseteq sc_{i+1} i=1..2 \}$
Operations	$Op=\{op_i \mid i=1..s, s\in \mathbf{N}\}$
Classification of operations	$\omega: Op \rightarrow SC$
Classification of objects	$\delta: O \rightarrow SC$
Confidentiality	$\psi: O \rightarrow S^n$
Clearance	$\phi: O \times S \rightarrow SC$
Authorization rules	$\forall s\in S, \forall o\in O, \Pi (s,o) =$ 0 if $s \in \psi(o)$ $\phi(o,s)$ if $s \notin \psi(o)$
Transition function	$\theta: Op \times O^n \times S \rightarrow O^m$, $(n, m \in \mathbf{N})$ $\theta(op, (o_1, o_2,..., o_N), s)$ is performed iff: a) $\omega (op) \leq \delta(o_i)$, and b) $\Pi(s, o_i) \neq 0$, and c) $\omega (op) \leq \Pi(s, o_i) \forall o_i, i=1,\ldots,n$

Subjects (S) is the collection of kinds of users who can access the hypermedia application exercising different permissions. Two kinds of subjects are considered: **roles** (R) and **teams** (T): a role is an organizational position or job function that appears in the application domain, as defined in RBAC models [9], whereas a team is an aggregation of roles defined to represent existing groups of interest (e.g. Member of a course on flooding), collaborative teams (e.g. 24H response team) or just to alleviate administrative tasks (e.g. system users). Roles and teams support composition mechanisms in order to deal with complex user structures. To gather the complexities of most organizations, hierarchies of roles can be defined as a Directed Acyclic Graph (DAG) where junior roles are specialized into more specific ones using a *is-a* relationship (generalization) involving inheritance. Teams are aggregations of roles and teams, a *whole-part* relationship that allows to refer to a number of subjects as a whole.

Users (U) are assigned one or more roles to be able to access the system through an **Assignment** function (A). A basic and static mechanism of **Separation of duties** is supported by means of a set of pairs of mutually exclusive roles. The **objects** (O) are the **nodes** (N) and **contents** (C) making up the hypermedia application, whilst links inherit the access rights defined for the nodes and contents they are embedded into. Nodes are abstract containers of information items, called contents. Thus, in order to see a link, a subject must be allowed to see all of its anchors (soruces and destinations), otherwise the link will not appear in any node accessed by such user. Two composition mechanisms are supported for nodes and contents: generalization, implying inheritance, and aggregation, used to refer to a group of objects as a whole. The object set takes the form of a DAG, by means of two partial orders defined by irreflexive, transitive and antisymmetric aggregation and generalization relationships. On the basis of the composition mechanism, the concept of **domain** is introduced to refer to the hierarchical structure defined from an object:

domain(o) = {o} $\cup$ domain(o') $\cup$ domain(o") $\forall$ o', o" | o'$\in$aggregatedBy(o) and o" $\in$generalizedBy(o)

$\forall$ o$\in$O, aggregatedBy(o)=list of objs aggregated by o,

$\forall$ o$\in$O, generalizedBy(o)=list of objs generalized by o

Thus, domains are used to refer to a number of objects, irrespective of how and where they are hold (as attributes of a relational table, as files or even as smaller components like tags). The domain concept is more powerful than the physical concept of directory used in some access models. The domain of the root node of a hypermedia system includes all the nodes in the system. A number of generic hypermedia **Operations** (Op) are supported for the different components of the hypermedia web system, such as *createNode*, *createContent*, *placeContentNode*, *createLink*, *activateLink* and so on. For a comprehensive list of operations see [15]. **Access categories** (Sc) are the kinds of access categories supported. A hypermedia web system is composed of a Basic Hyperdocument, accessed by all the users respecting certain access rules, and a number of Personal Hyperdocuments, that are private spaces only accessed and managed by their owners, who can be individual users or groups, represented by MARAH subject. To deal with these two concepts, three values of access categories, making up a partial order relationship (each one adds permissions to the previous), are considered: (1) Browsing, to retrieve information (nodes and contents) whether selecting links or using other means (maps, search engines, etc...); (2) Personalizing, that adds the ability to include personal elements (such as private contents, nodes or links), that is, the possibility of managing personalized spaces; and (3) Editing, that adds the ability to modify the hypermedia web system. These categories are used to assign each operation in Op the privileges needed to perform its tasks by means of the **classification of operations** (ω) function. For example, the *createNode* operation requires an Editing category, while *activateLink* is assigned a Browsing one. This classification is hard coded into the specification of the Labyrinth operations. Objects are assigned an access category to disable any authorization rule temporarily by using the **classification of objects** (δ). Thus, if during the update of a web site the security manager assigns a Browsing category to the hypermedia system, authorization rules of subjects will not be considered and no user will be able to modify any element of the system, irrespective of the permissions the subject is granted. Thus, since no authorizations have to be modified, no erroneous access rights can be exercised. Objects can be assigned negative ACLs where the subjects who are not granted any kind of access are held by means of the **confidentiality** relation (ψ). Positive authorizations are managed by means of the **clearance function** (ϕ) where a subject is granted an access category for an object. Here, the term clearance is not used in its classical meaning (security level assigned to a subject), but as a security level assigned to a subject for a specific object or domain to support context-dependent authorizations. The **authorization rule** (Π) gathers both relationships, so provides the manipulation abilities each subject will have in each domain. As MARAH is a model that can be used in different domains of application (e-learning, e-gov...), authorizations are defined using objects and generic operations, so that MARAH authorizations match the concept of permission in [3]. Domain-dependent operations (e.g. create course, solve exercise or access statistics in e-learning) are introduced at a higher level of abstraction (see the Functions in ADM), while MARAH offers generic operations (like *createNode*) applied in different domains (create a course, create a product…). Authorizations are propagated across the subject structure according to the following rules:

R1. *Direct propagation of authorization from junior roles*

$\forall\ s_1 \in S, \forall\ o \in O \mid s_1 \in \text{generalizedBy}(s_2), s_2 \in S \Rightarrow \Pi(s_1, o) = \Pi(s_2, o)$

R2. *Authorization overriding in senior roles*

$\forall\ s_1 \in S, \forall\ o \in O \mid s_1 \in \text{generalizedBy}(s_2), s_2 \in S, \Pi(s_1, o)=x \wedge \Pi(s_2, o)=y,\ x,y \in \{0\} \cup SC \Rightarrow \Pi(s_1, o)=x$

R3. *Authorization propagation in nested generalization relationships*

$\forall\ s_1 \in S, \forall\ o \in O \mid s_1 \in \text{generalizedBy}(s_2) \wedge s_2 \in \text{generalizedBy}(s_3)\ s_2, s_3 \in S \Rightarrow \Pi(s_1, o) = \Pi(s_2, o)$

R4. *Authorization propagation in parallel generalization relationships*

$\forall\ s \in S, \forall\ o \in O \mid s \in \text{generalizedBy}(s_i), s_i \in S, i=1..n \Rightarrow \Pi(s, o) = \Delta_1^n(\Pi(s_i, o))$

$$\Delta_i^n(\Pi(s,o)) = \begin{cases} 0, \text{ if } \Pi(s_i,o)==0,\ \forall i,\ i=1..n \\ \Pi(s_k,o) \mid \Pi(s_k,o) \geq \Pi(s_i,o),\ \forall i,\ i=1..n \end{cases}$$

R5. *Direct assignment of authorization for team members*

$\forall\ s_1 \in S, \forall\ o \in O \mid s_1 \in \text{aggregatedBy}(s_2), s_2 \in S \wedge \neg\exists\ \Pi(s_1, o) \Rightarrow \Pi(s_1, o) = \Pi(s_2, o)$

Finally, the **transition function** (θ) is responsible for guaranteeing that only safe operations are performed. This approach separates the policy enforcement point from the operation involved in the access decision, making the transition function independent of the action to be executed. Before its execution, every Labyrinth operation (Op) calls the transition function, so that programmers do not need to write policy enforcement code. MARAH is not a closed model; several extensions are foreseen, including a more complex SD mechanism, delegation, or temporal roles. However, it offers a powerful mechanism to specify access rules for hypermedia systems as it has been shown in projects like ARCE.

3.2 The ADM design meta-models

Based on the components of the MARAH meta-model, ADM offers a number of design models that are closer to the designers' perspective and can be integrated with other design views, so that a holistic design process can be faced. There are two kinds of ADM models depending on the abstraction level: conceptual models gather the features of the system in a very abstract way using types of entities that are instanced through detailed models. Table 2 lists all ADM models; those related with access modeling are highlighted and introduced below. For a comprehensive description of the method and its models see [11].

Table 2. The ADM design models

CONCEPTUAL DESIGN	
DESIGN MODEL	DEVELOPMENT CONCERN
Structural Diagram	Structural relationships among nodes
Navigation Diagram	Navigation paths and tools
Functional Specifications	**No-navigation services**
Internal Diagrams – Spatial Diagram	Node/content visualization area
Internal Diagrams – Timeline	Node/content evolution throughout a time interval
Attributes Catalogue	Properties and meta-data that can be used with different purposes
Events Catalogue	Behaviors that can be associated to functions
Users Diagram	**Expected types of roles and teams that can be used to support security rules or personalized/adaptive hypermedia**
Categorization Catalogue	**Security category for each object**
Authorization Rules	**Functions allowed to each role**
DETAILED DESIGN	
DESIGN MODEL	DEVELOPMENT CONCERN
Node Instances	Concrete instances of abstract nodes
Instanced Users Diagram	**Concrete instances of abstract roles and teams**
Specifications of Access Structures	Low-level specification of navigation aids
Detailed Specification of Functions	**Low-level specification of each function**
Detailed Internal Diagrams	Detailed information of nodes and contents
Users Allocation	**Specific users assigned to roles**
Presentation Specifications	Presentation properties for nodes and contents
Access Table	**Permissions expressed as rights to concrete contents, links and services**
EVALUATION	
MODEL	DEVELOPMENT CONCERN
Prototype	Interface mock-up
Evaluation Document	Report about the evaluation process
Conclusions Report	Conclusions to improve the system

Conceptual design of access needs consists of identifying types of roles and teams, functions to be performed in the system and finally the access rights. The functions that can be performed on the system are listed in the Functions Specifications where functions are decomposed into sub-functions as shown in Figure 2.

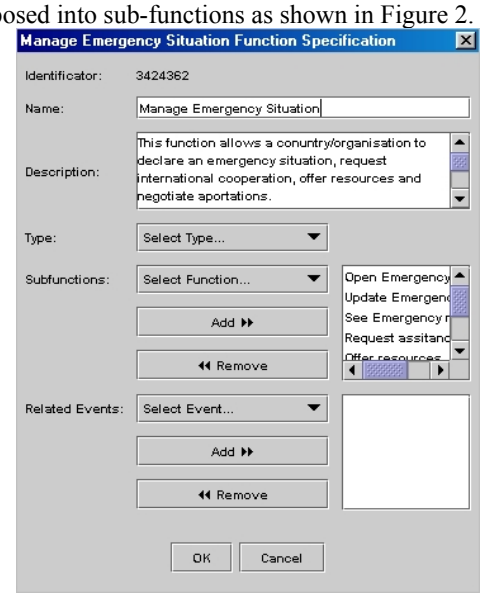

Figure 2: Functional Specifications for Manage Emergency ARCE function

The types of roles and teams are depicted in the Users Diagram where a DAG of conceptual subjects is defined as shown in Figure 3. High-level access rules (see figure 1) are established through the Authorization Rules (see figure 4). As it can be seen these high-level specifications are readable enough as to be discussed with stakeholders before the system is implemented so that errors and misunderstandings can be solved in design time. All these conceptual models are more specified during the Detailed Design whose models are mapped directly to MARAH components. Thus, conceptual teams and roles (like N4, for example) are replicated to create as many concrete roles and teams as needed (for example N4.Spain, N4.Argentina, N4. Portugal and so on) using the Instanced Users Diagram.

Functions are specified in terms of Labyrinth atomic operations which include the implementation of the Transition Function (see MARAH description) in the Detailed Specification of Functions. For example, managing the emergency situation will suppose to have an Editing category access for the emergency node what, in turn, will consist on having more items on the menu to manipulate the information. Concrete access rules are

gathered in the Access Catalogue. Moreover, some objects (nodes and contents) can be assigned specific access constraints using the Categorization Catalogue. For example, a node (page) can be blocked by setting its access category to browsing**.** Finally, concrete users are assigned the concrete roles they will be able to exercise in the system.

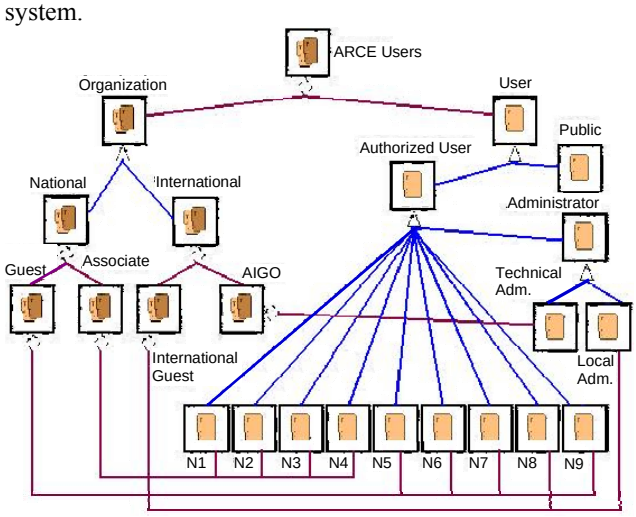

Figure 3: Users Diagram of ARCE

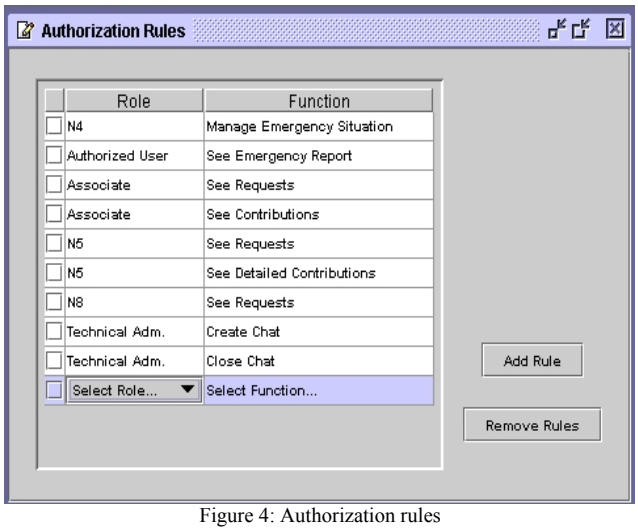

Figure 4: Authorization rules

4. THE ARCE WEB SYSTEM

ARCE is a WEMIS oriented towards enhancing the management of multinational disaster response within the scope of the Latin-American Association of Governmental Organisms of Civil Defense and Protection (AIGO), a multinational organism made up of representatives from 21 Latin-American countries. It is a platform to share updated and reliable information among the AIGO associates and other agents in order to orchestrate an integrated and efficient response, respecting the autonomies of each member. ARCE focuses on information integration regarding resource requests and offers, allowing the contributing countries to take into account the offers made by others. ARCE isn't expected to be used in emergency situations exclusively. It has two operation modes, routine and emergency, since if the WEMIS is not used on a daily basis it won't be used when an emergency or crisis situation happens.

4.1 ARCE Roles Specification

As it will be detailed in section 5, ARCE adopts a RBAC model to capture organizational structure of AIGO representatives. This task requires a good balance between the autonomy of each associate and the difficulty of managing and combining different structures. The need of coming up with a common structure for all associates, which preserves the required flexibility is accommodated by the inherent configurability of RBAC. Table 3 summarizes the roles identified in ARCE whose hierarchical structure is shown in fig. 4. The following issues are key in the role-based model adopted in ARCE:

1. Since a user can have several roles, different responsibilities can be carried out by the same person. This allows associates to decide how to assign their users to roles.
2. The notion of role is open, so that roles can be used to represent job positions, particular skills or even entire organizations that do not require further decomposition.
3. Roles can be used in an abstract way to define a general structure, which is instantiated or refined for each associate as needed. This allows to maintain a global coherence and to simplify access policy design at early development stages.

3.2 Routine mode of ARCE

In this operation mode ARCE offers a communication service and a news board. Communication is supported among the associates and other related institutions or external organizations (e.g. NGOs) as well as among the member of specific communities through messages and a chat. The **messages module** uses an information flow policy to distribute messages among the different agents. Information flow policies are aimed at ensuring that users only access the information for which they are authorized. For this purpose, both users and information have to be classified. In ARCE, users are classified using the roles in table 1 and information is categorized as strategic, operational, technical, general and public. Information flows from one role to another according to the rules in Figure 5.

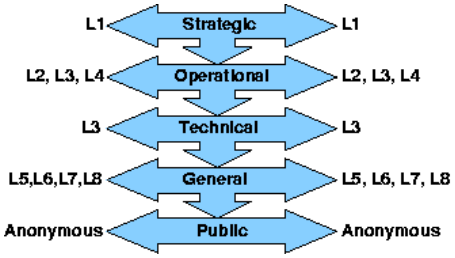

Figure 5: Information flow policy in ARCE

When a user is about to send a message, first she has to

Table 3. Roles in ARCE

Roles		Description
Associated Organism (One per each country of AIGO)	N1	Strategic level: General Director
	N2a	Operational level General Vice-Director
	N2b	Operational level: Chief of the Regional Organism
	N3a	Technical level: expert or technical
	N3b	Technical level: Expert in natural risks
	N3c	Technical level: Expert in technological risks
	N4a	Operational level: Operations Chief at the 24H Coordination Centre, Operations chief at the organism
	N4b	Operational level: Operator at the 24H Coordination Centre
National invited organism (Each associated organism can invite as many as wanted)	N5	Informative level: National agencies and organisms, which can invite users with same role.
	N6	Informative level: National agencies and organisms, which can't invite users.
	N7	Informative level: Non governmental organizations which can't invite users.
International invited organism (AIGO can invite as many as wanted)	N8	Supranational organizations (e.g. NATO or EU)
Information provider (invited by an associated organisms or AIGO)	N9	News provider
System administration	Technical admin	Maintenance operations concerning the whole system
	Local admin	Maintenance operations of a specific organism (users and organism's information)
Public	Anonymous	Any non authenticated user

decide the kind of information to be sent and then she will be able to select who is going to receive it from a list of potential targets. For instance, an L2 role will not be able to send strategic information although she could receive it from the upper level, and she can send Operational information to users holding an N2, N3 or N4 role. Note that a message sent to a role is received by all users assigned to the role. Communication service also includes a **chat module** used to set up forums on different issues concerning civil defence. Finally, there is a **news board** where several roles, including external organizations, can post news. For instance the National Geographic Institute of Spain and the CIIFEN (International Center for Research about "El Niño") provide news to keep informed the community and the society (see figure 6). For trustworthiness purposes, only authorized users can post news.

4.3 Emergency mode of ARCE

In this operation mode, countries affected by a emergency, crisis or disaster can manage the situation by informing the other associates, preparing a preliminary request for urgent

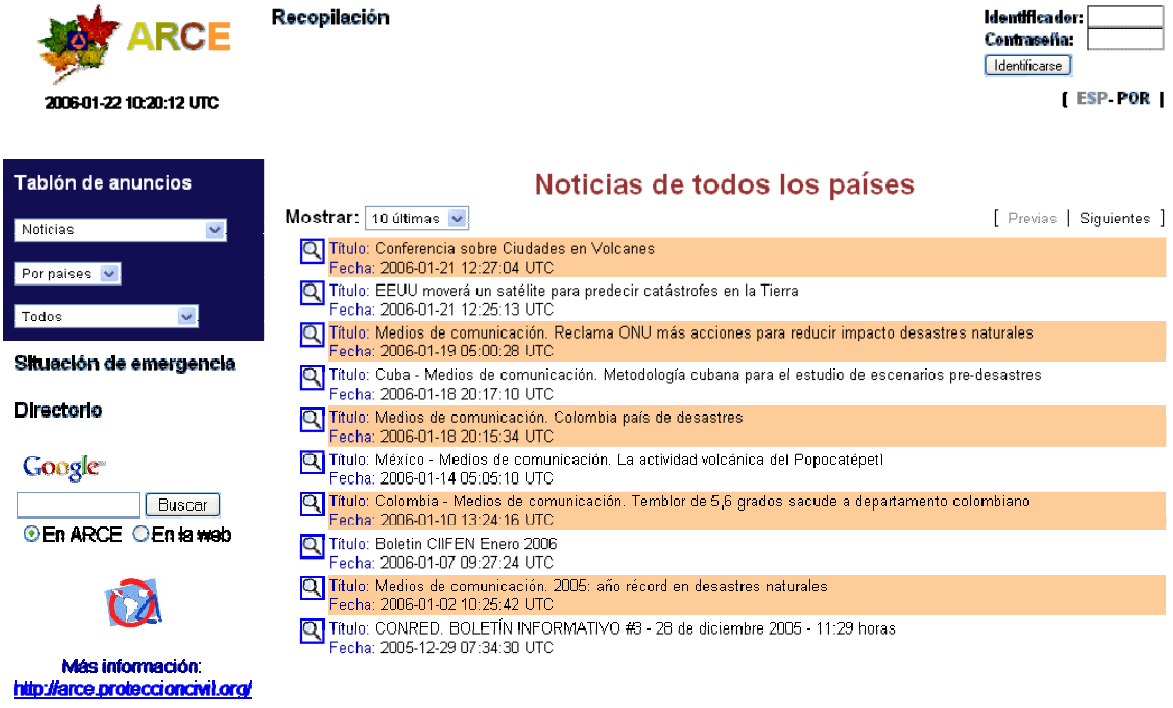

Figure 6: News in ARCE

resources, elaborating a more detailed request and coordinating the assistance offered by other countries. Since all the entities involved in an emergency, whether the owners or the assistance suppliers, have access to updated and reliable information about the real needs and how they are being solved, there are no problems of overlapping help. Indeed, before a supplier initiates the protocol to physically send any help its assistance has to be approved by the emergency owner. If assistance is required, the emergency owner can ask for resources (see Figure 7) which are classified using a multilingual catalogue of means and resources, that has been accepted by all AIGO associates and that also includes the SUMA (http://www.disaster-info.net/catalogo/English/dd/Ped/sumacat.htm) catalogue of humanitarian supplies. Whenever an assistance request is received, the rest of the associates are notified by e-mail, and eventually by Fax or SMS, so that they can access the system to see what the emergency owner is asking for and how they can help (see Figure 8). For each emergency, there is updated information on which resources are requested, what quantity was originally needed and how many items have been already supplied. Thus each associate can decide what to contribute taking into account what the others are doing. The assistance is always subdued to a negotiation process amongst the emergency owner and the assistance suppliers, so that suppliers cannot send anything till the emergency owner has validated explicitly the offer. The negotiation also allows the emergency owner to suggest changes to the offers sent by suppliers, which in turn may accept the changes, or suggest more ones. Eventually, the emergency owner can finalize the negotiation by accepting or denying the offer of a given associate. All the actions performed in the system can be audited using the historic mechanism which holds information what was done in past emergency situations managed with ARCE.

5. MODELING THE ARCE COMMUNITIES ACCESS NEEDS USING RBAC

One of the main features, and at the same time one of the main design challenges, is the diversity of responsibilities and kinds of users that make up the spectrum of potential ARCE stakeholders:

- On the one hand we have a number of governmental agencies that have complex and diverse organizational structures, so that our design has to meet the needs of all of them providing a common workplace but respecting their divergences. Creating an ad-hoc design for each of the 21 organizations is not an efficient solution in terms of maintainability, so that we need to specify a common structure that can be adapted to the needs of each of them in an easy way.
- On the other hand, there are a number of users (like invited organizations, ngos, journalists, anonymous users...) who can also access the system to keep themselves informed or even to contribute to mitigate a specific disaster.

In order to deal with different access rights the basic principles of RBAC have been adopted and, therefore, the rules governing the system access are specified using a hierarchy of roles. In particular as we said before, we have used the ADM web engineering method and its underlying security meta-model, MARAH.

ADM establishes a complete, systematic, iterative, flexible and user-centred development process that consists of three phases: Conceptual and Detailed Design phases address requirements design from different abstraction levels, and the Evaluation phase is based on prototypes and design models assessment. Each phase is decomposed into some activities, each of which generates various models [11]. In ARCE we applied the method in several cycles, each of which has followed a different approach. The assumption of this iterative process has been considered quite useful both for developers, who could test their ideas and solutions in a realistic situation, and by stakeholders, who have adopted a quite positive attitude due to their active involvement in the development process.

5.1 First cycle: analysis through rapid prototyping

At the beginning of the project, the basic goal was to establish a set of feasible requirements as well as to engage stakeholders in the development process. In this case, the ADM evaluation phase, with prototyping and evaluation as main activities, played a central role. This first cycle consisted of an iterative process of analysis and evaluation. Analysis was done using a typical working group technique, where some web engineers and people with technical knowledge in civil defence issues discussed the system services. Rapid throw-it-away prototypes were used to support requirements elicitation and validation through evaluation processes. Evaluations were chiefly oriented towards collecting and refining requirements and the method was an empirical evaluation of interface mock-ups. Particular emphasis was done on the study of different user roles and permissions.

5.2 Second cycle: producing a common design

The next step was to focus on technical solutions involving the stakeholders in this process. The idea was to deepen on the basic features and services of the system producing a common conceptual design that could be adapted later to the needs of each associate. In this second cycle, the basic activities are Conceptual Design and Evaluation of prototypes and design models.

At the Conceptual Design phase design solutions are expressed in terms of expected types of elements that will be translated afterwards into concrete entities in the Detailed Design Phase. In the ARCE project, the Conceptual Design offered a most useful choice to establish a coherent structure and function, since it provides a number of models readable enough to be discussed with stakeholders after a short explanation on the notation. Moreover, compared to the use of prototypes, Conceptual Design models hide details that can deviate the users attention to issues, such as colours, backgrounds and so on, which are not relevant when trying to define generic

Figure 7: A request from the point of view of an emergency owner

Figure 8: An assistance from the point of view of an assistance supplier

and abstract features of the system. This second cycle was again an iterative process devoted to refining both design models and requirements. Design was focused on:

- **System structure:** the *Structural Diagram* model captures the information structures of the application. ARCE structures support the management of emergency and the routine mode components. Figure 9 shows part of the Structural Diagram, where boxes represent information units which include contents (such as images or text) and other information units. For example, the "Emergency" unit contains a "Report", a set of "Assistance" offers (whether immediate or detailed) and a emergency "Closure" page.
- **Services:** the *Functional Specifications* model gathers the application functions offered to users. A function may be decomposed into smaller functions. Figure 3 shows the functional specification of the Manage Emergency function, which in turn is composed by some subfunctions such as "Update Report", "Create Request", and so on.
- **Access policy**: the *Users Diagram* and *Authorization*

Rules models are shown in figures 2 and 4 respectively and gather the roles structure (graphical representation of the elements in Table 1) and the high-level access rules (see figure 1).

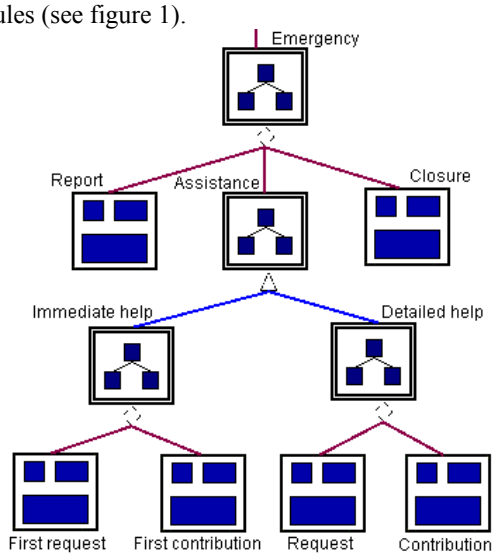

Figure 9: Excerpt of ARCE Structural Diagram

As said before and in order to have more expressive and powerful user's structures, MARAH introduces to the roles hierarchy the concept of team. A role is an organizational position or job function that appears in the application domain (e.g. the technical administrator of the system), whereas a team is an aggregation of roles defined to represent groups of interest (e.g. the participants of a chat on prevention of natural risks), collaborative teams (e.g. the International team in ARCE that aggregates international associations and other international communities that can be invited to provide assistance in a specific emergency situation) or just to alleviate administrative tasks (e.g. ARCE users). Roles and teams support composition mechanisms in order to deal with complex user structures. This way, one can define more general entities (such as "Organization" or "Authorized User" in figure 8) as well as more concrete ones (eg N1 role). As mentioned before, organization structure can be defined at different levels of abstraction so that it can be validated in the earlier phases of the development process. Access rules are assigned to roles and teams and they are propagated through the role hierarchy following a number of well-defined rules. Each user will be associated with one or more roles and not only the role assumed by a specific user will determine her capabilities to modify or browse the information but also the country she belongs to, since ARCE is implemented using context dependent roles. A context dependent role can be viewed as a role instance, made up of the name of the role (e.g N4) and the country for which the role is defined, so that in ARCE the context is defined as the location of the role, giving place to differentiated, concrete roles such as N4.Argentina and N4.Spain. Finally, roles are also the basis to establish an information flow policy that is used to push valuable information to each ARCE user (see Figure 5). One of the main benefits of this product is that we could establish a common roles structure that is latter adapted to the needs of each organization in the third cycle.

All these design models were assessed with the stakeholders to refine them. Sometimes such analytical evaluation gave place to changes in the prototypes and even in the requirements. Empirical evaluations of the prototypes were also performed in a number of simulation exercises with a view to assess the system utility. These exercises, where real users took part, made possible to improve the system.

5.3 Third cycle: adapting designs to each organization

Next step was to keep on refining design and validating if it suits the specific organizational features of each associate. In this third cycle, Analysis activities were practically anecdotal and the ADM Detailed Design came into the scene to define concrete instances of some conceptual models. All the models of the Conceptual Design were developed and refined and Evaluation continued as a basic activity to assess both design models and prototypes.

During the Detailed Design phase entities and services are fully specified. Such specific elements can be identified in a declarative way (e.g. using an identifier, url o uri) or in a procedural one (e.g. by means of scripts or database queries). Indeed, most instances of nodes and contents in ARCE were defined in a procedural way by accessing a PostgreSQL database. The access policy is also specified in a more detailed way in terms of concrete subjects and objects, that is, using low-level permissions (see Figure 1) by means of two models: the *Access Table* and the *Users Assignment*. The Access Table gathers the access rights each subject instance can exercise with each object instance and its values are computed automatically from the high-level design products (see figures 2 and 3) by granting each role or team the minimum clearances needed to execute each atomic operation making up the high-level function. For example, if role N4 can execute the function "Create emergency report", she has to be able to create a new Report as well all to modify all the fields included in such page (Emergency Location, Type, Damages…). Specific users are associated roles through the *Users Allocation* model. Thus, users will be able to exercise the abilities specified for the roles they belong to according to the principles of RBAC models. A user can be associated to more than one role to increase flexibility.

6. CONCLUSIONS: ON THE BENEFITS OF RBAC TO MODEL WEMIS COMMUNITIES

Emergency and crisis management involves many different stakeholders who make up virtual communities with different characteristics, needs and responsibilities. A useful WEMIS should cope with this diversity providing each user the right information and services in the right way and in the right moment. In this work we have focused this problem exclusively from the perspective of the access

policy that will make possible to specify the access rules in a flexible and efficient way. The multiple HCI principles that have to be considered to deliver the information and services in the right way go beyond the goals of the work here presented. In particular, we advocated for the use of the RBAC paradigm whose benefits to specify organizational policies have been highlighted elsewhere [4, 5]. Our approach consist of applying an RBAC meta-model within the context of a development methodology for web systems, so that access requirements can be specified at different levels of abstraction and in a integrated way, that is relating them with other requirements (functional or non-functional). From this experience we can draw a number of benefits both from the use of an RBAC models and for its application within an engineering method that are summarised in the following paragraphs.

RBAC is flexible and expressive enough to capture the access needs of different kinds of WEMIS communities. Since the WEMIS is accessed by different agents the access policy has to be carefully designed to support enough roles as to provide a certain degree of autonomy and control while maintaining reliability and efficiency. To avoid improper access and modification of data ARCE relies upon the use RBAC policies and authentication mechanisms, so that only authorized users can modify the information provided by the system. Moreover, the information flow policy ensures that messages received from the system are trustworthy, as far as only authorized users can send messages to the users who require or need that information. Though in an emergency situation it can not be predicted who will undertake a specific role [2], it is also obvious that a user cannot assume a role for which she isn't trained before since no organization will allow any kind of anarchical procedure where a user decides on her own to assume a role for which she isn't prepared nor authorized. In such a scenario, the use of roles, hierarchies of roles, teams and the possibility of assigning each user different roles provides a powerful and flexible specification mechanism to gather the needs of many different kinds of communities in terms of access rules. For this purpose, the use of constraints for user allocation (where the assignment of a user to a role depends on a certain condition) is also quite useful. For example, the number of times a specific user has played a role in simulation exercises might be used to dynamically determine whether she's allowed or not to assume such role in an emergency.

Moreover, in a multinational environment the access policy has to be flexible enough to support different organizational policies while maintain a global coherence and consistence. In our case, this requirement was fulfilled thanks to the application of the web engineering method. During the Conceptual Design stage we established a common structure and access policy that can be discussed by all the involved stakeholders. Such common design was adapted to each organization during the Detailed Design making use of the expressiveness underlying user assignment so that in each organization each users was assigned as many roles as needed. During the ARCE operation users are not aware of the roles they are playing, they just authenticate and are provided with access to all the functions they can exercise according to the roles they have been assigned. Moreover, user assignment can be performed at any moment so that roles can be granted in operation time. In this sense, the model-based approach should be extended to automatically support the delegation of roles (as suggested in [6]) since actually it has to be done through the local systems manager.

RBAC can improve the communication among designers and stakeholders. RBAC abstractions (basically the concepts of role, hierarchies of roles, operation, authorization and user allocation) make possible to specify the access policy using a language that is close to the stakeholders since it is based on organizational entities.

In our case the application of RBAC at different levels of abstraction was a good option to specify the policy in an incremental way and validate it with the stakeholders through analytical and empirical evaluations. The discussions between the ARCE stakeholders and designers to define the Users Diagram (the model where roles and teams are identified) and the Authorization Rules (where roles and teams are allocated the functions they can perform) were particularly useful to clarify the different responsibilities that should be considered by the WEMIS. Indeed, the design models became intellectual tools to think about how organizations were doing things and even to detect problems in the organizational structure (such as duplication of efforts, erroneous assignment of responsibilities, etc.)

Moreover, as we did translate this policy into the prototype we could validate and improve these products in a number of simulation exercises where several associates took part in an emergency practice.

However there are a number of extensions that can be done in the model-based approach here described to increase its expressiveness including mechanisms to allow for the delegation of roles, the definition of temporal roles or the dynamic separation of duties. We are actually applying this approach in other WEMIS such as SIGAME (www.sigame.es).

7. REFERENCES

[1] Dykstra, E. H. Towards an International System Model in Emergency Management: information communication and coordination in emergency management- public and private sector approaches in different countries and systems. Communication. Public Entity Risk Institute (PERI), September 22-26, 2003.

[2] Turoff, M. Chumer, B. Van de Walle and X. Yao, "The design of a dynamic emergency response management information system", Journal of Information Technology Theory and Applications (2004) 5:4, 1-36.

[3] ANSI INCITS 359-2004, American National Standard for Information Technology; Role Based Access Control. 2004.

[4] Barkley, J., Ferraiolo, D. and Radack, S. An Introduction to Role-Based Access Control. NIST CSL Bulletin, December 1995.

[5] Ferraiolo, D.F., Barkley, J.F. and Kuhn, D.R.: A Role-Based Access

Control Model and Reference Implementation within a Corporate Intranet. ACM Trans. on Information and Systems Security, 2(1), February (1999), 34-64.

[6] Zhu, H. and Zhou, M.C., "Role-Based Collaboration and its Kernel Mechanisms", IEEE Trans. on Systems, Man and Cybernetics, Part C, vol. 36, no. 4, July 2006, pp. 578-589.

[7] Díaz, P., Sanz, D., Montero, S. and Aedo, I. Integrating Access Policies into the Development Process of Hypermedia Web Systems. In Web and Information Systems Security. Eds. Ferrari, E. and. Thuraisingham, B. Idea Group Inc. pp. 149-172. 2006.

[8] Brose, G., Koch, M. And Löhr, K.P. Integrating Security Policy Design into the Software Development Process. Technical Report B–01–06. Freie Universität Berlin, Nov. 13. 2001.

[9] Devanbu, P.T. and Stubblebine, S. (2000). Software engineering for security: a roadmap. The Future of Software Engineering. Finkelstein, A. ed. ACM Press.

[10] Jürjens, J. (2002). UMLsec: Extending UML for Secure Systems Development. UML 2002, Dresden, Sept. 30 - Oct. 4, 2002, LNCS 2460, Springer. 412-425.

[11] Díaz, P., Montero, S. and Aedo, I. Modelling Hypermedia and Web applications: the Ariadne Development Method. Information Systems 30(8): 649-673. Dec. 2005.

[12] Kropp, B. and Gallaher, M. P. Access to cost savings: Role-based access control systems can save organizations time and money. Information Security Maganize (April). 2001.

[13] Gallaher, M.P., O'Connor, A.C. and Kropp, B. The economic impact of Role-based Access Control. Planning Report 02-1. National Institute of Standards & Technology. March, 2002.

[14] Aedo, I., Díaz, P. and Montero, S. A methodological approach for hypermedia security modelling. *Information and Software Technology*, 45(5), 2003, pp. 229–239.

[15] Díaz, P., Aedo, I. and Panetsos, F. Modelling the dynamic behavior of hypermedia applications. IEEE Transactions on Software Engineering, 27(6):550-572, June 2001.

[16] "Model Driven Architecture", Document number ormsc/2001-07-01, Architecture Board ORMSC 2001

Ignacio Aedo has a degree and a doctorate both in Computer Science from the Polytechnic University of Madrid. Since 1992, she has been working in the Carlos III University where she is currently a Full Professor in the Computer Science Department. His research areas include access modeling, emergency management systems, human computer interaction and e-learning. He is co-author of several articles and books concerning his research activities.

Paloma Díaz received a degree and a doctorate both in Computer Science from the Polytechnic University of Madrid. Since 1992, she has been working in the Carlos III University where she is currently a Full Professor in the Computer Science Department and is the head of the DEI research group. She has been mainly researching in hypermedia /web engineering, access control modeling, e-learning and emergency information systems. She is co-author of several articles and books, member of ACM and senior member of the IEEE.

Daniel Sanz obtained the CS Engineering Degree at Carlos III University in 2002. Then he started to work as assistant teacher at this University and as PhD Student in the CS Doctorate Program. His research topics are the application of RBAC to hypertexts, both at theoretical as well as practical/technological levels. He also is interested in the integration of access control in hypermedia development methods.