Goal-Oriented Access Control Model for Ambient Assisted Living

Fabio Massacci, Viet Hung Nguyen

Goal-oriented Access Control Model for Ambient Assisted Living

Fabio Massacci and Viet Hung Nguyen

University of Trento, Italy
{fabio.massacci, vhnguyen}@unitn.it

Abstract. Ambient assisted living is a new interdisciplinary field aiming at supporting senior citizens in their home by means of embedded technologies. This domain offer an interesting challenge for providing dependability and security in a privacy-respecting way: in order to provide services in an emergency we cannot monitor on a second-by-second base a senior citizen. Beside being immoral, it would be illegal (at least in Europe). At the same time if we do not get notified of an emergency, the entire system would be useless.
In this paper we present an access control model for this domain that extends RBAC with the notion of organizational model, goals and dependencies. In this model we can associate permission to the objectives that have been assigned to the users of the system and solve the trade-off between security and dependability.

1 Introduction

Ambient assisted living (AAL) [25, 24, 7] is a home environment enhanced with embedded technologies (sensors, cameras, and similar electronics devices) in order to support elderly people's daily tasks. This raises numerous challenges related not only to technology i.e., interaction between human and smart devices [32, 28], but also to the safety and security [22] of the human living in such environments.

From a privacy and security perspective, two kinds of challenges are identified:

- *Dependability*: The life of the elderly people will be at risk if important data are not accessible at the right time;
- *Privacy*: Private data are being delegated from system to system so the privacy of the person is at risk as well.

To protect data privacy, when sensitive data are being processed, the access should be justified by a certain purpose requiring the disclosure of the data. So the authorization to access certain resources is not only based on the entitlement to use a resource, but also on the purpose for which the resources are being used. Such principle is summarized with the phrase: *no purpose, no data*.

In the domain of database this is well understood. In fact, the protection of customer privacy is a legal requirement that any enterprise information system

has to fulfill and enforce. Not surprisingly, many research efforts have proposed new privacy-aware technologies. Among them, Hippocratic databases offer mechanisms for enforcing privacy rules in database systems for inter-organizational business processes [1]. In [20], Massacci et al. extend those mechanisms in order to implement hierarchical purposes, distributed authorizations and minimal disclosure supporting the business processes of virtual organizations. The proposed framework uses a goal-oriented approach to analyze privacy policies of the enterprises involved in a business process.

In contrast, we do not find an equally large number of comprehensive security solutions in the domain of Ambient Assisted Living addressing the issue of purpose. Indeed the solution on the US side is the exact opposite of what EU legislation would mandate: collect all data and the identify sophisticated rules for access control [31]. We could define sum this policy as *collect and protect*. Beside being illegal in the EU this approach has two major scalability problems: at first the complexity of managing the security policies and second and foremost the complexity of managing the actual data.

In general the collection of sensitive data without a specific purpose is illegal in Europe. Data about video surveillance is subject to even stricter regulations. Of course, a company in charge of a smart-home maintenance might try to cover itself by collecting blanket privacy give-aways by its customers but such attempts would be struck down and heavily sanctioned by the privacy commissioner if legally challenged.

As an example in Italy (which has a weaker legislation than Germany) distance monitoring of workers is strictly forbidden and patient monitoring in hospitals is only allowed in special wards (rianimation) and anyhow subject to preliminary approval (Garante della privacy ruling in 2004 [13]):

"Video surveillance equipment should only be activated if other measures (alarm systems, sensors, etc.) are considered to be insufficient and/or unfeasible following a careful analysis. [...] Supervision of medical facilities and monitoring of patients hospitalised in certain departments and/or units such as resuscitation units should be limited to the cases in which this is absolutely indispensable on account of the sensitive nature of many data to be possibly collected in this way, by limiting the scope of surveillance to certain premises and well-defined time ranges. "

As it is immediately clear that if even in a resuscitation unit you cannot run a 24/7 monitoring by humans the idea of remote day-by-day monitoring in a home is far beyond what is legally possible, no matter how much consent forms you collect (in the same way that you can't collect signatures of people accepting to be sold in slavery).

Consider just the issue of video monitoring. Even the local provider for elderly and public housing in Trento, a sparsely populated Italian province, has well over 1000 houses, scattered among valleys and mountains (which explains why they are interested in AAL solutions). The cost for getting connectivity, storage, and security protection measures for the wealth of sensor and video streaming of all

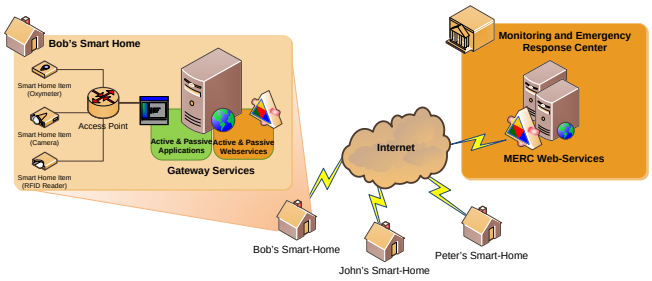


Fig. 1. E-health system infrastructure.

collectible data would largely exceed the cost of hiring a personal nurse for each of the elderly people in question.

1.1 Contributions of this paper

We introduce a formal access control model extending RBAC which is called Goal-oriented role based access control (GoRBAC for short). And based on it, we are aiming at limiting the issued authorizations to the permissions needed to fulfil the current goal of the involved actors or sub-systems.

This access control model has been fully implemented and demonstrated in a real smart-home. We present here only the formal aspects of the model and refer to [21] for the details of the demonstration scenario. A video representing the real system is also available on the web (`http:\\www.disi.unitn.it\~massacci`).

In the rest of the paper we present our case study on Ambient Assisted Living (§2). Then we present the formal notion of Organizational Model (§3) and notion of Goal-Oriented Access Control (§4) and its dynamics. Finally we discuss related work (§5) and conclude the paper (§6).

2 The Ambient Assisted Living Scenario

For the demonstration purpose, in our work we consider a typical *eHealth* application where an old man living alone in his smart-house. The house is embedded with different smart-devices (oximeter, camera, and so on) to monitor the person 24/7. It is also able to detect whether he is endangered and sends an emergency alert to the Monitoring and Emergency Response Center (MERC).

In particular, there are three scenarios of the *eHealth* application are taking into account as follows.

Normal Operation: It is the normal situations with usual daily activities.

Emergency: In the second one, the patient feels dizzy and falls down in the kitchen. Moreover, the oximeter reports that his heart rate is too high. According predefined detection rules, it is recognized as an emergency. The smart-home security manager sends an alert message to MERC. MERC access to smart-home (whose security manager has changed the right of access

following a suitable pattern) to retrieve his medical data and the snapshot at the falling time as well. When the emergency is confirmed, MERC setups a rescue team and sends it to smart-home. When the rescue team arrives, the smart-home's WSN detects their identity and the security manager send it to MERC for authorization. The rescue team then are correctly authorized by MERC, afterward the smart-home security manager sends a one-time password to MERC who in turn forward it to the rescue team i.e., by SMS. The rescue team use this password to open the smart-home.

Social worker: In the final scenario, the patient is recovered, but he still need some medicine treatment. The medicine are delivered to smart-home by a social worker from the hospital.

No.	*Scenario*	*Security challenges*
1	Normal Operation	The patient should be monitored 24/7 even if one monitoring device fail. The collected information should not be accessed from outside even the MERC. No one could not enter the house with out the patient's agreement.
2	Emergency	MERC should be able to access the sensors' data. The rescue team are allowed to open the door and accessed medical data for a proper pre-treatment. These permissions are temporarily granted, and should be revoked when the emergency ends.
3	Social worker	The social worker can open the door if the patient could not do this, but the social worker should not be able to access patient's medical information when he is in the house.

Table 1. Security challenges in the scenarios.

These scenarios show a challenge to the smart-home security manager: MERC should be able to collect medical data from his smart house (and also other smart houses). In the meanwhile, to comply with the privacy law, the security manager should not not let data out until it serves some purposes. These security challenges are summarized in Table 1. The basic infrastructure of a such system is depicted in Figure 1.

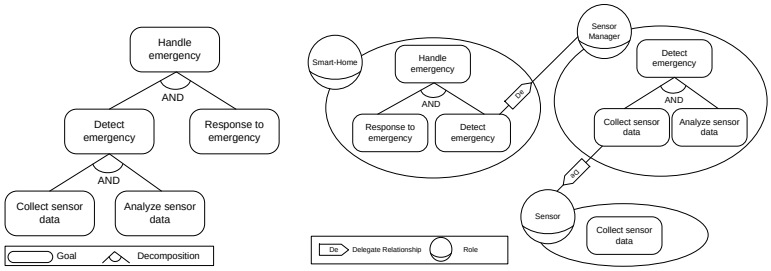
3 A Goal-Oriented Organizational Model

The organizational model proposed here is based on the security-requirements engineering methodologies presented in [14] for socio-technical systems. The original model has been simplified by restricting it to functional goals and adapting it to the security notion of roles instead of using the notion of actors. Simplification was necessary also because the original work was focussing on requirements engineering where a rich set of construct is a feature while here we need to make extremely fast run-time decisions.

A *goal model* consists of a set of *goals* and their relationships. Goals are recursively decomposed until they arrive to concrete *operational goals* (*operations* for short) which could be directly assigned to human or software components to in order to be achieved. We consider a simple way to decompose goals which is the *means-end* decomposition. This relationship shows that the *end goal* is obtained if the *means goals* are achieved.

Example 1. The objective of the MERC is to handle emergency which is can be refined into the two subgoals detect emergency and response to emergency. The detection of emergencies can be further refined in another possible ways to detect the urgent situations as Collect sensor data, Analyze sensor data.

The goal model is graphically represented in Figure 2(a), in which each goal is denoted as a round rectangle. The goal model can be formally defined as follow.



(a) Goal model (b) Organizational model

Fig. 2. A portion of goal model (a) and corresponding organizational model (b) of the case study

Definition 1. *A goal model is a triplet $\langle G, Dc, OP \rangle$, in which G: is a set of goals representing stakeholder objectives and requirements. $Dc \subseteq G \times 2^G$: is a set of one-to-many Means-End-decompositions which constitutes an acyclic relations. $OP \subset G$: is a set of operational goals which can be fulfilled directly by actors.*

In comparison with the original goal model in [14] the notion of AND-decomposition of goals has been collapsed into the means end-decomposition, and we do not explicitly represent OR-decomposition as it is captured by different means-end decomposition of the same goal.

While this might be strictly less precise (as decomposition it is not the same as means-end), it greatly simplifies the cognitive overhead of policy writers and the run-time efficiency of enforcement.

An *organizational model* is constructed by adding to a goal model, a set of roles, the hierarchy among roles and the assignments of goals to roles. Loosely speaking a role is an abstract characterization of the behavior of a socio-technical actor within the domain. Loosely speaking GoRBAC roles corresponds to RBAC

roles with goals on top. The assignment schemes include the goals-to-roles assignments and the goals' decompositions to roles assignments.

Example 2. We add three roles to the goal model described in Figure 2(a): Smart-Home, Sensor Manager, Sensor. The goal-to-role assignment and delegation relationship are depicted in Figure 2(b).

Definition 2. *An organizational model, $\mathcal{M}$, is tuple of $\langle \mathcal{M}^{\mathrm{G}}, R, \mathcal{A}_{G\times R}, \mathcal{A}_{Dc\times R}, De\rangle$, where $\mathcal{M}^{\mathrm{G}}$: is a goal model. R: is a set of role. $\mathcal{A}_{G\times R} \subseteq G \times R$: is an assignment of goals to roles. $\mathcal{A}_{Dc\times R} \subseteq Dc \times R$: is an assignment of goal decompositions to roles. $De \subseteq \{R \times G \times R\}$: is set of dependency relations in which one role can depend on another role to fulfill certain goal.*

The assignment of goals to roles is an obligation, that means the agent playing that role must satisfy all its assigned goals. A goal can be assigned to different roles and vice versa. Once a role is in charge of a certain goal, it can satisfy this goal if this goal is a concrete operation, or decompose this goal into other subgoals, or delegate it to another role. Different roles might choose different way to fulfill and thus refine the goals. This explains why we needed to associate the particular assignment of goal decompositions to roles.

Goals, goal decompositions assigned to a particular role and available delegation relations that originate from this role are called *role model*.

Definition 3. *Given a role r, a* role model *is the tuple $\langle \mathcal{A}^r_{G\times R}, \mathcal{A}^r_{Dc\times R}, De^r\rangle$ where $\mathcal{A}^r_{G\times R}, \mathcal{A}^r_{Dc\times R}$ and De^r are, respectively, set of goals, set of decompositions and set of delegation assigned to r.*

Goals can be decomposed in many ways but we must be sure that agents in charge of their fulfillment can actually do something in order to achieve them. In other words, the human or the the system playing a role can decide either to satisfy the goal itself, or delegate some subtask to other role.

Definition 4. *An organizational model, $\mathcal{M}$and a role r and a goal g assigned to r the goal g is* actionable *for r if*

- *$g \in OP$ is a concrete operation, or*
- *there exists $\langle g, S_G\rangle \in \mathcal{A}^r_{Dc\times R}$ and for all goals $g' \in S_G$, either g' is actionable or there exists a role r' and a delegation $\langle r, g', r'\rangle \in De^r$ such that g' is actionable for r'.*

Example 3. In Figure 2(b), the role configuration of Sensor Manager includes three goals, one decomposition and one delegation. Analyze sensor data is a concrete operation performed by Sensor Manager, and Collect sensor data is delegated to Sensor. Thus, these two goals are actionable, and Detect emergency is actionable as well. Therefore, this role configuration is actionable.

Since roles are not physical entities, goals are actually satisfied by *agents* (or principals) which are actors with concrete, physical manifestation such as human individuals or machines.

Remark 1. We prefer to use the notion of agents rather then the common term *users* because the intuitive understanding of users in this scenario is that they corresponds to human beings. Agents in this setting can be either human or software agents in the same sense that Alice and Bob in security protocols are often just dramatis personae for actual software processes running the protocols.

The configuration also defines the assignment of agents to each role.

Definition 5. *Given an organizational model $\mathcal{M}$, an organizational configuration $\mathcal{M}^c$ is a tuple of $\langle \mathcal{A}^c_{G\times R}, \mathcal{A}^c_{Dc\times R}, De^c, A, \mathcal{A}_{A\times R}\rangle$, where $\mathcal{A}^c_{G\times R}$ is an assignment of goals to roles, $\mathcal{A}^c_{Dc\times R}$ is an assignment of decompositions to roles, De^c is set of delegation among actors, A is a set of agents, $\mathcal{A}_{A\times R}$ is an assignment of agents to roles. The following properties hold:*

1. $\mathcal{A}^c_{G\times R} \in \mathcal{A}_{G\times R}$, $\mathcal{A}^c_{Dc\times R} \in \mathcal{A}_{Dc\times R}$, $De^c \in De$, $\mathcal{A}_{A\times R} \subseteq A \times R$;
2. *For each role $r \in R$, r has an actionable configuration within $\mathcal{M}^c$.*

In other words we require that the current assignments of roles to agents is such that all goals can be fulfilled. So an agent playing a certain role r might delegate something to another role r' but the system must be sure that there is actually some agent that can play the latter role r'.

This is only a necessary and not a sufficient condition for success: operations might fail in practice or other agents might fail to deliver. Upon notification of failures or successful achievements, the run-time system must make sure that the appropriate configurations are selected.

An organizational model can have many configurations. The one currently considered by the run-time system is the *active configuration*.

4 Goal oriented RBAC

So far we have only defined the functional goals of the system and not yet introduced the notion of permissions. The main idea behind GoRBAC is to strengthen (and weaken at the same time) a traditional RBAC access control decision using the organizational model. In fact, the grant of a permission to access an object is not an end per se but it is a mean to achieve a goal.

As in traditional RBAC, we want to ensure that only authorized users are allowed to access the resources. However, different strategies can be used for defining when and how these authorizations are issued.

- *Privacy*: the main issue for the privacy strategy is to ensure that the *privacy-critical resources* are accessed only by authorized agents when needed. This strategy implements the principle "no purpose no data". The definition 6 clarifies the meaning of the purpose of an operation in our model.
- *Dependability*: in a dependability context, the system aims to maximize the probability of successful fulfillment of the critical goals. The derived permissions are generated once the user is authorized the fulfillment the top-level goal. In particular, if a service have different decompositions, we derive permissions for all of them in order to increase the availability of the service.

Definition 6. *Given an organizational model $\mathcal{M}$and its active configuration $\mathcal{M}^{\mathrm{C}}$, the purpose of an operation is a set of goals which satisfy follows:*

$$Purpose(op) = \{op\} \cup \{g \in G \,|\exists g_1 \in Purpose(op), \exists S_G \subset 2^G, \exists r \in R \\ \langle \langle g_1, S_G \rangle, r \rangle \in \mathcal{A}'_{Dc \times R} \wedge g \in S_G\}$$

We give a simplified version of the traditional RBAC as defined in [26].

Definition 7. *RBAC model is a tuple $\langle U, R, OP, P, \mathcal{A}_{U\times R}, \mathcal{A}_{P\times R}\rangle$ where A is a set of agents, R is set of roles, $\mathcal{A}_{A\times R}$ is a agent-to-role assignment, $P \subseteq OP$ is a set of permissible operations (or permissions). $\mathcal{A}_{P\times R} \subseteq P \times R$ is a many-to-many permission-to-role assignment relation.*

The traditional RBAC distinguishes between operations and objects, and then pairs them into permissions. In the AAL setting such distinction is not always useful. There are operations that requires simultaneous access to a number of objects and can be better understood by users if explained at this level of details.

Example 4. Access to patient data requires to have access to the positioning information of the camera and the oximeter readings. Turning-on the camera is a simple operation from the point of view of the user but at the software level is a complex operation that requires access to a number of objects starting from the IP address to the camera.

This definition can be extended as usual with hierarchies and sessions(see[26]).

The set of authorizations which constraints whether an user u is able to do an operation *op* is defined as follows:

$$\mathcal{A}_{RBAC} = \{\langle u, op\rangle \,|\exists r \in R.\, \langle u, r\rangle \in \mathcal{A}_{A\times R} \ \wedge \ \langle op, r\rangle \in \mathcal{A}_{P\times R}\} \tag{1}$$

We define the GoRBAC as an extension of RBAC as follows.

Definition 8. *A GoRBAC Model is a tuple $\langle RBAC, \mathcal{M}, \mathcal{G}, \mathcal{M}^{\mathrm{C}}, \mathcal{P}\rangle$, where $RBAC$ is the RBAC model, $\mathcal{M}$ is the organizational model, $\mathcal{G} \subset G$ is a set of critical goals and $\mathcal{P} \subset P$ is a set of privacy sensitives permissions, $\mathcal{M}^{\mathrm{C}}$ is the active configuration of the organizational model.*

The following property must also hold: For each $\langle g, r\rangle \in \mathcal{A}_{G\times R}$, if an operation op has a purpose g then it is assigned to r.

The permission of performing an operation *op* is granted to an agent a if:

1. if the operation *op* serves for the satisfaction of a critical goal a is fulfilling.
2. else if the secure object is privacy sensitive then this operation should serves for the satisfaction of a goal which a is fulfilling and a is authorized to access this object regard to the security policy in RBAC model.
3. else a is authorized regard to the security policy in RBAC model.

To this end, beside the GoRBAC model, the runtime security management maintains a record describing the active agents and their fulfilling goals at runtime. We call this record *runtime configuration* defined as follow:

Event	**Actions**
$Add\_agent(a)$	–
$Activate\_role(a, r)$	**CHECK** $\langle a, r\rangle \in \mathcal{A}^c_{A\times R}$ **DO** $\mathcal{A}^*_{A\times R} \leftarrow \mathcal{A}^*_{A\times R} \cup \{\langle a, r\rangle\}$
$Activate\_goal(a, g)$	**CHECK** $\exists r \in R. \langle g, r\rangle \in \mathcal{A}^c_{G\times R} \wedge \langle g, r\rangle$ *is able to activate* $\wedge$ $\{a, r\} \in \mathcal{A}^*_{A\times R}$ **DO** $\mathcal{A}^*_{G\times A} \leftarrow \mathcal{A}^*_{G\times A} \cup \{\langle g, a\rangle\}$
$Delegate(a_1, g, a_2)$	**CHECK** $\exists r_1, r_2 \in R. \langle a_1, r_1\rangle \in \mathcal{A}^*_{A\times R} \wedge \langle a_2, r_2\rangle \in \mathcal{A}^*_{A\times R} \wedge \langle g, a_1\rangle \in \mathcal{A}^*_{G\times A} \wedge \langle r_1, g, r_2\rangle \in De'$ **DO** $\mathcal{A}^*_{G\times A} \leftarrow \mathcal{A}^*_{G\times A} \cup \{\langle g, a_2\rangle\}$ $De^* \leftarrow De^* \cup \{\langle a_1, g, a_2\rangle\}$
$Goal\_Fulfilled(a, g)$	**CHECK** $\langle g, a\rangle \in \mathcal{A}^*_{G\times A}$ **DO** Deactivate g **DO** Propagate the fulfillment to parent goals related to g and update accordingly their fulfilment and active status.
$Goal\_Failed(a, g)$	**CHECK** $\langle g, a\rangle \in \mathcal{A}^*_{G\times A}$ **DO** Deactivate g **DO** Check the fulfillment status of the other parent goals of g in the active configuration and update accordingly their fulfilment and active status.
$Deactivate\_Role(a, r)$	**CHECK** $\langle a, r\rangle \in \mathcal{A}^*_{A\times R}$ **DO** $\mathcal{A}^*_{A\times R} \leftarrow \mathcal{A}^*_{A\times R} \setminus \{\langle a, r\rangle\}$ $\forall g \in G. \langle g, a\rangle \in \mathcal{A}^*_{G\times A} \wedge \langle g, r\rangle \in \mathcal{A}'_{G\times R}$, deactivate the child goals of g in the active configuration.
$Undelegate(a_1, g, a_2)$	**CHECK** $\langle a_1, g, a_2\rangle \in De^* \wedge \langle g, a_1\rangle \in \mathcal{A}^*_{G\times A} \wedge \langle g, a_2\rangle \in \mathcal{A}^*_{G\times A}$ **DO** $De^* \leftarrow De^* \setminus \{\langle a_1, g, a_2\rangle\}$ **DO** Deactivate g and its child goals in the active configuration. **DO** Unfulfill the child goals of g in the active configuration.

Table 2. List of basic events updating the runtime configuration

Definition 9. *Given an active configuration, $\mathcal{M}^c$, of a system. The runtime configuration of the system is defined as a triplet $\left\langle \mathcal{A}^*_{A\times R}, \mathcal{A}^*_{G\times A}, De^* \right\rangle$, where*

- $\mathcal{A}^*_{A\times R} \subseteq \mathcal{A}^c_{A\times R}$ *is an active agent-to-role assignment,*
- $\mathcal{A}^*_{G\times A} \subseteq \mathcal{A}^c_{G\times A}$ *is an active goal-to-agent assignment,*
- $De^* = A \times G \times A$ *is a set of active delegation relationships among agents. The following property should be valid.*

$$\langle a_1, g, a_2\rangle \in De^* \rightarrow \exists r_1, r_2 \in R. \{\langle a_1, r_1\rangle, \langle a_2, r_2\rangle\} \subseteq \mathcal{A}^c_{A\times R} \wedge \langle r_1, g, r_2\rangle \in De^c$$

The runtime security management maintains the runtime configuration and modifies it with respect to events. The Table 2 presents the basic events that the security manager takes into account for updating the runtime configuration.

The security request $\langle a, op\rangle$ is granted if and only if:

$$\langle a, op\rangle \text{ is granted } \begin{cases} \text{if } \exists g \in \mathcal{G}. g \in Purpose(op) \text{ and } \langle g, a\rangle \in \mathcal{A}^*_{G\times A} \\ \text{elseif } ; op \in \mathcal{P} \text{ and } \langle g, a\rangle \in \mathcal{A}^*_{G\times A} \text{ and } \langle a, op\rangle \in \mathcal{A}_{RBAC} \\ \text{elseif } \langle a, op\rangle \in \mathcal{A}_{RBAC} \end{cases}$$

In this way the fulfillment of critical goals always override whatever setting of permission needed to accomplish the task at hand. This is an absolute requirements for emergency services. For example, in many medical authorization system, a red button "Night shift", when only few doctors are present, is present to override any normal authorization process. Obviously, logging procedures might be put in place to monitor such events.

Example 5. In an emergency context, authenticating the rescue team against the smart home is considered as a critical goal. We can imagine different authentication mechanisms providing different levels of robustness. The most robust mechanism could be defined as the default one but if we are missing some resources to fulfill it, the system will activate any other available mechanism.

At the same time, if the data is privacy sensitive you do not want it to be accessed unless there is some purpose that has been actually assigned to the user requesting the permission.

Example 6. The medical data of the patient is considered as privacy sensitive resource. Therefore, the access to it is regulated by the "no purpose no data" principle. The social worker is allowed to access it only if it is playing rescue team member role during an emergency context.

For normal authorizations we fall back to the standard RBAC authorization. At this point a genuine conflict might arise: the user might be assigned by the organization a goal which he cannot fulfill. This happens frequently in daily life. However, since the goal is not critical for the organization, we can as well afford the time to let the user go back to the system administrator and solve the problem with the required care.

Example 7. During an ordinary check on the patient status, only his doctor is allowed to access his data

5 Related Works

In our case, the security requirements of the system concern the access to the resource available in AAL environment. Traditionally, the access control policy is defined as a list of permissions that is statically defined at design time [11, 26, 16, 4, 23]. For RBAC, once a role is activated at runtime, all related permissions are also activated. Using a hierarchy of roles, we can limit the set of permissions that are activated at the same time but still any subject S playing a role R is entitled to use all the related permissions no matter if it needs them or not for its current activities.

Moyer and Abamad have proposed a Generalized access control model (G-RBAC) [23]. GRBAC introduces new concepts such as subject roles, object roles and environment roles. Subject roles are like traditional RBAC roles, object roles abstract the various properties of objects, and environment roles capture environmental information, such as time of day. All these meta-information about

objects and subjects introduced through these new concepts increase the expressiveness of the RBAC model, allow a fine access control decision and a more flexible access control scheme.

RBAC constraints [2, 5, 8] are essentially used to enforce higher level organization security policy such as LP and SoD principles . These constraints can be related to user-role assignment, role-permission assignment or to some runtime context conditions [10, 18]. In this last case, even if the user is entitled to access a certain resource, the actual authorization is given only after checking the related constraints. From an administration point of view, the usage of constraints is fundamental for enforcing higher level security or privacy requirements but it also increases complexity of maintenance related activities.

OrBAC [15] and Multi-OrBAC [16]. OrBAC introduces context as a new entity to specify the circumstances in which the organization grants permissions on objects. In Multi-OrBAC, each role and permission is valid in a specific organization. This model is more adapted to distributed and heterogeneous systems.

dRBAC (Distributed RBAC)[12] has been proposed as an access control framework for Dynamic Coalition Environments. It is intended to be decentralized trust-management and access-control mechanism for systems that span multiple administrative domains.

All these frameworks are interesting and appropriate in their application domains. However, none of them compare issued permissions against the real needs of the user from a functional point of view. This issue is fully delegated to administrators off-line. At runtime, the system checks if the request satisfies more or less sophisticated conditions of some stored permissions in order to grant the access. So any user can dispose of all their privileges even if they are not needed for the current activity they are performing.

Active security models for access control are those defining the permissions at workflows and operations level [29]. They defined Conceptual Foundations for a Model of operation-based Authorizations. The permissions in these models are associated to the activity of the system and this constraint is expressed in terms of an association between access operation and workflow activity.

T-RBAC (Temporal RBAC) has been introduced by Bertino et a. in [4]. It addresses the dynamic aspects related to periodic activations and deactivations of roles, and temporal dependencies among these actions actions.

For systems dealing with privacy sensitive data, different privacy frameworks and languages have been proposed to specify the privacy requirements and enforce them at runtime. Among work centered on the notion of purpose, LeFevre et al. [19] enhance Hippocratic databases with mechanisms enforcing queries to respect privacy policies stated by an enterprise and customer preferences. In essence, they propose to enforce the minimal disclosure principle by providing mechanisms to data owners that control who can access their personal data and for which purpose.

To support the negotiation of private information, the World Wide Web Consortium (W3C) proposed the Platform for Privacy Preferences (P3P) [9]. This standard provides mechanisms that allow customers to check web site privacy

policies before they disclose their personal data to the site. Another mechanism for negotiation is presented by Tumer et al. [30]. Enterprises specify which information is mandatory for achieving a service and which is optional, while customers specify the type of access for each part of their personal information

Mechanisms for enforcements are proposed by Karjoth et al. [3, 17]. The Enterprise Privacy Authorization Language (EPAL) [3] enables an enterprise to exactly formalize the privacy policies that shall be enforced within the enterprise itself. However, these proposals do not provide mechanisms for enforcing the minimal disclosure principle. In Byun et al. [6], the Role-Based Access Control model is extended by introducing the notion of purpose and a purpose management model. Similarly to our approach, they introduce purpose hierarchies in order to reason on access control. However, their hierarchies are based on the principles of generalization and specialization and are not expressive enough to support complex strategies defined by enterprises.

A policy itself may be sensitive because from the analysis of the disclosed policies an unauthorized user may infer sensitive information. Following this observation, some approaches propose to protect not only personal information, but also policies themselves [27].

We would like to evaluate GoRBAC against other RBAC based access control frameworks that have been cited in this section. We analyze the pros and cons of the cited RBAC extensions against two criteria :1) the least privilege principle and 2) AC policy management.

- Least Privilege principle: the comparison is based on the more or less constraints introduced by the new model with regard to the RBAc model.

  Example 8. For example, GoRBAC add an additional condition to be verified before grating the access. In fact, the model states that having the permission to access a resource, is a necessary but not sufficient condition. In addition the access request should be justified by the current responsibilities assigned to the requester. Thus GoRBAC enforces more the least privilege compared to RBAC.

- AC Policy management: the comparison is based on the complexity of specifying the policy rules and the granularity of the policy with regard to the real system operations.

  Example 9. In GoRBAC, the access control policy is specified at organizational level and it is based on the responsibilities assigned to the different roles inside the organization. The fine-grained access control policy related to every operations and every object in the system is derived automatically. Thus, we are clearly facilitating the tasks for the security administrators.

6 Conclusions

To sum-up this paper we have presented a novel access control model, GoRBAC, which take into account the purpose of operations. The model is based on the

notion of organizational model in order to implement the notion of "no purpose, no data" behind data access. To verify to model in experiment, we also developed a prototype [21] implementing the case study discussed in section 2. In that work, we deployed the prototype in the real environment, Smart-Home at Trento, and conducted the experiments with the scenarios presented in Section2.

References

1. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases. In *Proc. of VLDB'02*, pages 143–154, 2002.
2. G.-J. Ahn and R. Sandhu. Role-based authorization constraints specification. *TISSEC*, 3(4):207–226, 2000.
3. M. Backes, B. Pfitzmann, and M. Schunter. A Toolkit for Managing Enterprise Privacy Policies. In *Proc. of ESORICS'03*, volume 2808 of *LNCS*, pages 162–180. Springer-Verlag, 2003.
4. E. Bertino, P. A. Bonatti, and E. Ferrari. TRBAC: A temporal role-based access control model. *TISSEC*, 4(3):191–233, 2001.
5. E. Bertino, E. Ferrari, and V. Atluri. The specification and enforcement of authorization constraints in workflow management systems. *TISSEC*, 2(1):65–104, 1999.
6. J.-W. Byun, E. Bertino, and N. Li. Purpose Based Access Control of Complex Data for Privacy Protection. In *Proc. of SACMAT'05*, pages 102–110. ACM Press, 2005.
7. D. J. Cook and S. K. Das. How smart are our environments? an updated look at the state of the art. *Pervasive Mob. Comput.*, 3(2):53–73, 2007.
8. J. Crampton. Specifying and enforcing constraints in role-based access control. In *Proc. of SACMAT'03*, pages 43–50. ACM Press, 2003.
9. L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, Apr. 2002.
10. M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca. Geo-rbac: A spatially aware rbac. *ACM Trans. Inf. Syst. Secur.*, 10(1):2, 2007.
11. D. Ferraiolo and D. Kuhn. Role Based Access Control. In *15th National Computer Security Conference*, 1992.
12. E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti. dRBAC: distributed role-based access control for dynamic coalition environments. In *Proc. of ICDCS'02*, pages 411–420. IEEE Press, 2002.
13. Garante per la protezione dei dati personali. Video surveilance - the general provision adopted by the garante. Official Bullettin 49, April 2004. http://www.garanteprivacy.it/garante/doc.jsp?ID=1116810.
14. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Requirements Engineering for Trust Management: Model, Methodology, and Reasoning. *Int. J. of Inform. Sec.*, 5(4):257–274, 2006.
15. A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benfrhat, F. Cuppeens, Y., A. Mige, C. S. e, and G. Trouessin. Organization Based Access Control. In *4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003)*, 2003.
16. A. A. E. Kalam and Y. Deswarte. Multi-OrBAC: a New Access Control Model for Distributed, Heterogeneous and Collaborative Systems. In *8th IEEE International Symposium on Systems and Information Security*, 2006.

17. G. Karjoth, M. Schunter, and M. Waidner. Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data. In *Proc. of PET'02*, volume 2482 of *LNCS*, pages 69–84. Springer-Verlag, 2002.
18. D. Kulkarni and A. Tripathi. Context-aware role-based access control in pervasive computing systems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 113–122, New York, NY, USA, 2008. ACM.
19. K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. J. DeWitt. Limiting Disclosure in Hippocratic Databases. In *Proc. of VLDB'04*, pages 108–119. Morgan Kaufmann, 2004.
20. F. Massacci, J. Mylopoulos, and N. Zannone. Hierarchical Hippocratic Databases with Minimal Disclosure for Virtual Organizations. *VLDBJ*, 15(4):370–387, 2006.
21. F. Massacci, V. H. Nguyen, and A. Saidane. No purpose, no data: Goal-oriented access control for ambient assisted living. In *Security and Privacy in Medical and Home-Care Systems. In associated with CCS'09*, 2009. to appear.
22. S. Moncrieff, S. Venkatesh, and G. West. Privacy and the access of information in a smart house environment. In *Proc. 15th of MULTIMEDIA '07*, pages 671–680, New York, NY, USA, 2007. ACM.
23. M. Moyer and M. Abamad. Generalized role-based access control. In *ICDCS '01: Proceedings of the The 21st International Conference on Distributed Computing Systems*, page 391, Washington, DC, USA, 2001. IEEE Computer Society.
24. J. Nehmer, M. Becker, A. Karshmer, and R. Lamm. Living assistance systems: an ambient intelligence approach. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 43–50, New York, NY, USA, 2006. ACM.
25. H. Pigot, A. Mayers, and S. Giroux. The intelligent habitat and everyday life activity support. In *5th Int. Conf. on Simulations in Biomedicine*, pages 507–516, 2003.
26. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
27. K. E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting Privacy during On-line Trust Negotiation. In *Proc. of PET'02*, volume 2482 of *LNCS*, pages 129–143. Springer-Verlag, 2002.
28. M. A. Stelios, A. D. Nick, M. T. Effie, K. M. Dimitris, and S. C. A. Thomopoulos. An indoor localization platform for ambient assisted living using uwb. In *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 178–182, New York, NY, USA, 2008. ACM.
29. R. K. Thomas and R. S. Sandhu. Task-based authorization controls (tbac): A family of models for active and enterprise-oriented autorization management. In *Proc. of the IFIP TC11 WG11.3 11th Int. Conf. on Database Security*, pages 166–181, London, UK, UK, 1998. Chapman & Hall, Ltd.
30. A. Tumer, A. Dogac, and H. Toroslu. A Semantic based Privacy Framework for Web Services. In *Proc. of ESSW'03*, 2003.
31. Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B. K. Alshebli, M. Caccamno, C. A. Gunter, E. L. Gunter, J. Hou, K. Karahalios, and L. Sha. 1-living: An open system architecture for assisted living. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2006.
32. D. Xie, T. Yan, D. Ganesan, and A. Hanson. Design and implementation of a dual-camera wireless sensor network for object retrieval. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 469–480, Washington, DC, USA, 2008. IEEE Computer Society.