No Purpose, No Data: Goal-oriented Access Control for Ambient Assisted Living [*]

Fabio Massacci
DISI - University of Trento
fabio.massacci@unitn.it

Viet Hung Nguyen
DISI - University of Trento
vhnguyen@disi.unitn.it

Ayda Saidane
DISI - University of Trento
ayda.saidane@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 don't get notified of an emergency the entire system would be useless.

In this paper we present an access control model, the security architecture and the running implementation for ambient assisted living in smart-home. The model is based on the notion of organizational model in order to implement the notion of "no purpose, no data" behind data access. The detail of our prototype is presented in the video [1].

1. INTRODUCTION

Ambient assisted living (AAL) [7, 6, 2] 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 [10, 8], but also to the safety and security [5] of the human living in such environments.

We have two kinds of challenges:

- *Dependability*: The life of people will be at risk if important data is not accessible at the right time;
- *Privacy*: Private data is 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 [4], 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. This leaves us in a catch 22 situation when facing the solution proposed by current access control models or by US colleagues [9].

Contributions We aim at defining an access control model that implements the privacy principle "no purpose no data" as the extreme limit for "Least Privilege" (LP) principle. We propose in this paper a new goal oriented access control framework aiming at limiting the issued authorizations to the needed permissions to fulfil the functional requirements of the system. In fact, at a certain time the only permissions that are given by the system are those related to the goals currently being fulfilled.

In the rest of the paper we present our case study on Ambient Assisted Living (§2). Then we describe the notion of Organizational Model in §3 and our novel Goal-Oriented Access Control (GoRBAC) in §4. Next, our GoRBAC architecture and prototype are summarized in §5, §6, respectively. Finally, the section §7 concludes our work.

2. AMBIENT ASSISTED LIVING

Let's 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 man 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).

When MERC receive an alert, they can remotely access his medical data, including the cameras, to have precise re-

9[*]This work is partial supported by the EU committee with the grant of IST-TP-SERENITY and IST-FET-SECURE CHANGE.

9[1]http://www.dit.unitn.it/%7emassacci/Download/SERENITY-AVI.avi

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

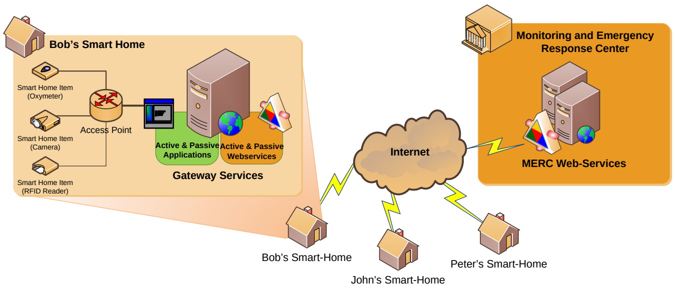
Bob's Smart Home
Smart Home Item (Oxymeter)
Smart Home Item (Camera)
Smart Home Item (RFID Reader)
Access Point
Active & Passive Applications
Active & Passive Webservices
Gateway Services
Monitoring and Emergency Response Center
MERC Web-Services
Internet
Bob's Smart-Home
John's Smart-Home
Peter's Smart-Home

Figure 1: E-health system infrastructure.

actions. If necessary, a rescue team is sent to support the patient. When the rescue team arrive, if the patient cannot open the door (e.g., he is unconscious), some authorizations will be given, and the rescue team can enter the house.

To the monitoring responsibility, MERC should be able to collect medical data from his smart house (and also other smart houses). However, regard to the privacy law, the house does not let data out until it serves some purposes.

The infrastructure of a such system is depicted in figure 1. On the smart home side, we envisage the following services:

- Active and Passive Services: are in charge of communicating with MERC. The communication is basically two way interrogation for exchange data. These services are also called Gateway Services.
- Active & Passive Applications: are bridge between Gateway services and smart home entities such as Sensors (WSN AP), Video Cameras and RFIDs.

3. A GOAL-ORIENTED ORGANIZATIONAL MODEL

At first we introduce a definition of organizational model based on the notion of goals. The model is based on the security-requirements engineering methodologies presented in [3] for socio-technical systems. We have simplified the model restricting it to functional goals and adapting it to roles instead of using the notion of actors. We will not present formal details due to lack of space, but we give examples from the formalization.

An *organizational model* includes a set of roles, a set of goals and a goal-to-role assignment scheme which details which goal belongs to which actor. The model also captures the dependencies among goals, actors. Goals can be decomposed into subgoals from the very high abstract-level to lower ones that are obligations of human actors or software components. A goal can have more than one way of decomposition, but at runtime only one is applied at a specific period. When being assigned goal, a role can continue decomposing this goal, or fulfilling it (if leaf goal), or delegating it to another role. These are called goal's configurations and the selected configuration is called *active configuration*.

Example 1. *Consider a portion of the abovementioned eHealth application's organizational model, the* smart-home *has a goal "handle emergency", and to achieve this objective, the smart-home divides it into four other subgoals namely "Detect emergency", "Response to emergency", "Show patient status", and "Support rescue team". The smart-home delegates the obligation of "Detect emergency" to the* Sensor Manager*. The sensor manager in turn separates this goal again into "Get sensor events" and "Detect emergency from sensor events". The former goal then is delegated to each sensor (e.g., camera, oximeter), meanwhile the later is fulfilled by the sensor manager.*

Our MinimumCost algorithm analyzes the organizational model to compute an optimal runtime organization configuration by selecting configurations such that all top goals are reachable from leaf operations with the minimum cost of goal satisfaction. The notion cost of satisfaction determines the necessary effort to satisfy a goal.

3.1 Dynamics of an organizational model

At runtime, the system organizational structure is not static but continuously evolving. However, this evolution is not random since we have considered and analyzed the important events affecting the security, privacy and dependability of the system.

We define **Potential Goal Model** as the role-level organizational model including all the possible configuration for fulfilling a goal and also all the dependencies between roles.

A runtime, we need to generate the initial **Active Goal Model (AGM)** as the actual object-level organizational model including all the actors, their dependencies and their actual goals. The AGM includes only the object level as we consider that in the real system we have only agents playing roles while the roles are abstract concepts used for management purposes. The AGM includes a set of active agents, and assignment themes of agent-to-role, agent-to-role and delegation between agent-to-agent.

A transition between a certain system state to an other one is feasible if needed modifications satisfy the constraints:

- The Potential Goal Model allows the modification.
- For role activation, there is no separation of duties constraint forbidding the activation
- For goal activation, the goal should belong to active configuration of a top level goal assigned to the user.

The dynamics of the goal models are described through the events that affect the organizational structure of the system and the actions that need to be taken by the organizational structure manager to update the model. Any modifications affecting the system goal model will have an impact on the issued authorizations related to the dependability critical and privacy sensitive goals.

We have two possible regimes:

In the **Reactive mode**, the system is acting as observer. Its main role is to ensure that the current active model corresponds to the real system. When a goal is fulfilled or failed, the system propagates its fulfillment to other goals in the active configuration related to g.

In the **Pro-active mode**, the system is acting as a manager. It maintains the active model updates and may suggest to users the optimal way for fulfilling certain goals. For dependability purposes, the system can have an important role during failure recovery process. In fact, it can suggest the alternative configurations for fulfilling critical goals or anticipating the need of goal constrained authorization that may be necessary to fulfill them.

The system starts by applying algorithm MinimumCost to compute a cost-optimal configuration. Then it monitors the satisfaction status of goals. When a goal is fulfilled, the system applies the same logic as it does in Reactive-mode. When a goal g fails, the system updates the status of the given goal, and computes another optimal configuration.

4. GOAL ORIENTED RBAC

Traditionally, the access control policy is defined as a list of permissions that is statically defined at design time. For RBAC model, once a role is activated at runtime, all the related permissions are also activated. 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.

The main idea behind GoRBAC is to limit the definition of the permissions to goal level and to drive and refine these high level permissions until operations and objects level. We constrain the access control decision using the system goal model describing the organizational model and specifying the different goals associated to a role and the different objects and operations needed for fulfilling these goals. 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.

For secure systems, 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".

dependability In a *dependability context*, the system aims at maximizing 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. Particularly, if a service have different decompositions, we derive permissions for all of them in order to increase the availability of the service.

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.

At the same time, if the data is privacy sensitive, user could not access it unless there are some purposes actually assigned to the user.

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.

At run-time, in order to avoid computing the operation needed to fulfill goals again and again, **Goal constrained assignment** for accessing the resources available in the system or executing particular operations on them, can be automatically derived from the goal assignment and the organizational model.

In this way, the goal-constrained permissions are only valid during the top level goal fulfillment according to the current organizational configuration.

5. GENERAL ARCHITECTURE

The reference architecture of the GoRBAC runtime enforcement framework, which is illustrated in figure 2, consists of:

GoRBAC Policy Generator It is an off-line administration tool, that allow the administrator to add classical permissions when needed, define the initial system goal model used for the initialization of the access control policy. The initial access control policy contains both RBAC basic permissions and goal constrained permissions related to privacy sensitive and dependability critical goals.

Runtime Organizational Structure Manager (OSM) It is an important part of the framework. The OSM gathers information about the system and its environment, filter the relevant data for updating the organizational model of the system and transfer the update model to the *GoRBAC manager*. The OSM plays the roles of *ProActive Manager* and *ReActive Manager* (see section 3.1) to react with organizational events such as a goal fulfills or fails. According to the arrival events and the operating mode, the OSM computes an optimal configuration (*ProActive* mode).

Access Control Manager This the central component taking the decision of granting or denying the access to the resources. It uses the current *organizational structure* model and the classic permission that might have been added by the administrator, for generating the *derived permissions* as explained previously.

The figure 2 represents the components needed for initialization, runtime mantainance and enforcement of the security policy. A monitoring service is needed to maintain a coherent view of the actual organizational structure of the system to update system goal model. The information that are expected from the monitoring service are related only to the agent-level. In particular, it provides information about which agents are joining and leaving the system, the roles they are playing and the actual goals they are fulfilling.

6. AAL PROTOTYPE DEPLOYMENT

The smart-home prototype is responsible for monitoring the patient and sending alerts to the MERC when necessary. It receives data from the different monitoring devices (oximeter, camera, for instance) and process it in order to

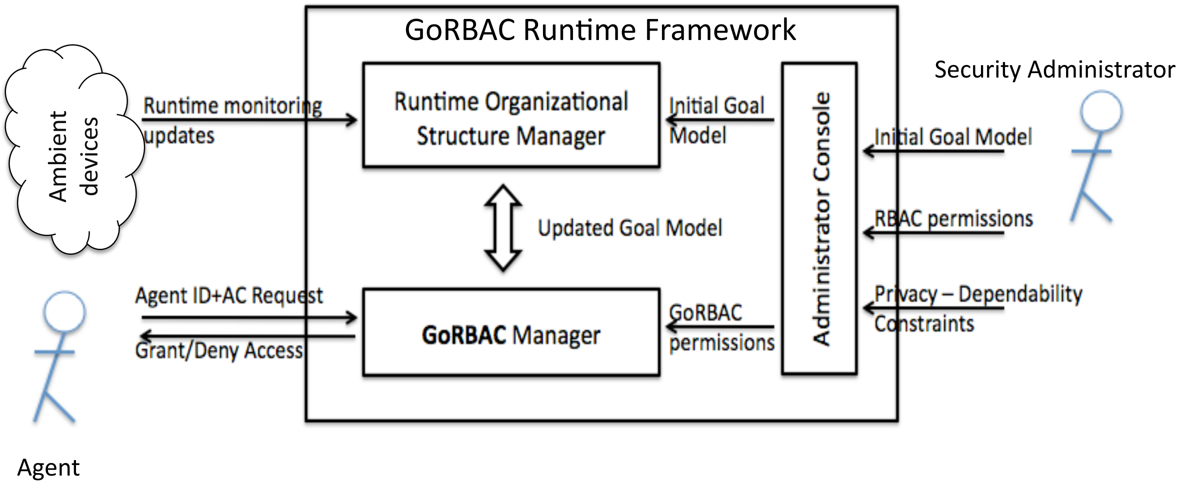

Figure 2: Runtime enforcement framework for GoRBAC

detect any emergency. We deployed the application in a real ambient assisted environment consisting in an IETA smart house (Figure 3). The Province of Trento has decided in 2004 to financially contribute to the installation of smart home technology in elderly people's flats of the ITEA Public Building Institute, including subsidies for both safety devices and personal aid devices up to a certain amount.

The general architecture of the prototype is composed of 4 components: AAL environment, AAL application, Security framework, and MERC server. The AAL application and the security framework were deployed on the same server : a XEON 4Core E5405 2.00Ghz, 4GB RAM DDR2, Dual Lan Gigabit. For the communication between the MERC and the smart house, we used AXIS2.

The prototype was developed to be autonomic and adaptive. It should be able to detect any changes in the organizational structure of the environment and react to the different events in particular those related to security, privacy and dependability. The execution of the prototype is explained more detail in the video which could be found at the url mentioned in the abstract.

AAL Environment It includes the smart house and the different devices. These components are related essentially to the monitoring and configuration of the environment. In the prototype, we deployed: *i*) *2 Cameras* (AXIS 212 PTZ) used for monitoring the patient status and detect fall down event. The controller associated to this device was developed in Visual C++ (.NET Framework 3.0). *ii*) *1 Oximeter* (Nonin 4100 Bluetooth) used for monitoring the heart rate the oxygen in the blood of the patient. The controller associated to this device was developed in Java 1.6.10. *iii*) *4 Motes* (CrossBow TelosB (IEEE 802.15.4)) used for identifying the human agents visiting the smart house. The controller associated to these devices is developed in NesC language.

Device handlers are mediators between the physical devices and the Event Server. They collect and process data, and then generate events to the Event Server.

AAL Application This category is composed of 3 components running all on the same server. *i*) The *Event Server* receives and forwards events from Device Handlers to the Event Manager for later processing. *ii*) The *Event Manager* is in charge of instantiation organization context into the OSM, and interacting with the Access Control Manager and the GoRBAC Authorization. Beside, the Event Manager provides an interface allowing the MERC system to access patient medical data in case of emergency. *iii*) The *Emergency Detector* is a rule-based detector that considers events from Event Manager to know whether the patient is in danger. The detection rule likes a determined-finite automata whose transitions are based on events.

GoRBAC runtime framework The GoRBAC runtime framework consists of four components : Administration Console, OSM, GoRBAC Manager, and Pattern Repository. The first three components are mentioned in the previous section. In this prototype, we implements OSM in *ReActive Mode* since the organizational model is simple and clearly defined. Though working in Reactive mode, the OSM is able to give suggestion for the fulfilment of critical goals. For instance, the "Get sensor events" goal is crucial for the emergency detection process; it is therefore delegated to more than one sensor for the reliability and dependency. To give such suggestion, each goal is marked with appropriate security and dependability requirements, and the OSM then looks for the most suitable *pattern* managed by the *Pattern Repository* according to the active environment settings.

7. CONCLUSION

In this paper, we have presented a novel access control model, called 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. By which, the permission is granted to a user if the predefined security policy allows this (static view) and the user is going to fulfill a certain purpose (dynamic view). Base on this work, we have developed an e-Health application prototype that monitors patients' activities 24/7 to determine they are endangered or not and to send an emergency request to MERC for help.

8. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases. In *Proc. of VLDB'02*, pages 143–154, 2002.

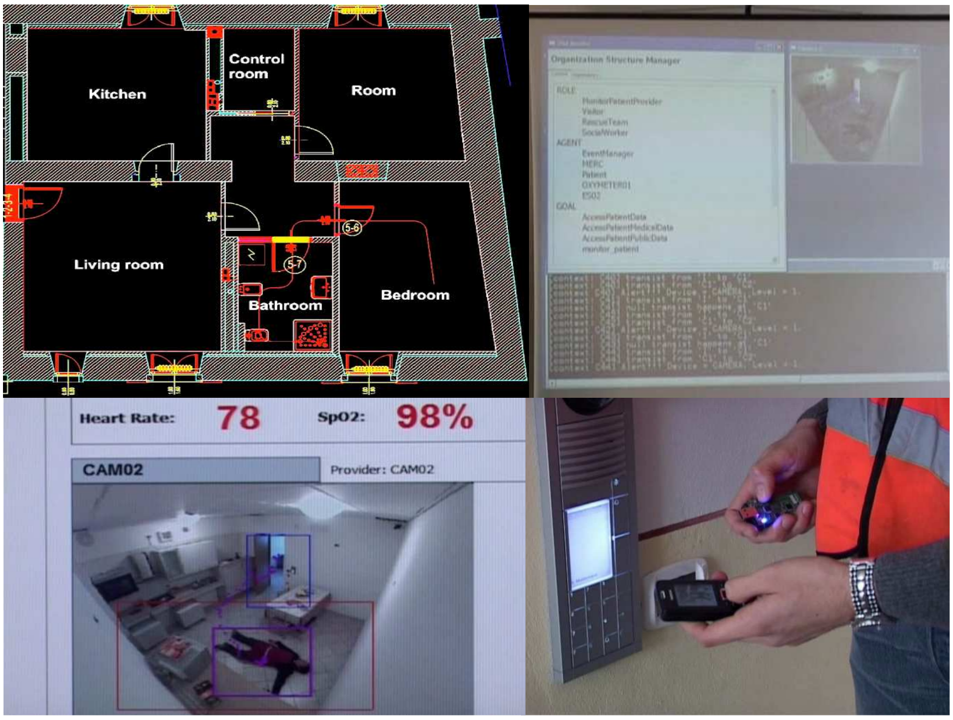

Figure 3: Deployment of the prototype in the ITEA Smart House

[2] 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.

[3] 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.

[4] F. Massacci, J. Mylopoulos, and N. Zannone. Hierarchical Hippocratic Databases with Minimal Disclosure for Virtual Organizations. *VLDBJ*, 15(4):370–387, 2006.

[5] S. Moncrieff, S. Venkatesh, and G. West. Privacy and the access of information in a smart house environment. In *Prof. of 15th Intl. Conf. on MULTIMEDIA '07*, pages 671–680, 2007.

[6] J. Nehmer, M. Becker, A. Karshmer, and R. Lamm. Living assistance systems: an ambient intelligence approach. In *Proc. of 28th Intl. Conf. on ICSE '06*, pages 43–50, 2006.

[7] H. Pigot, A. Mayers, and S. Giroux. The intelligent habitat and everyday life activity support. In *Prof. of 5th Int. Conf. on Simulations in Biomedicine*, pages 507–516, 2003.

[8] 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 *Prof. of 6th Intl. Conf. on MoMM '08*, pages 178–182, 2008.

[9] 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 *Prof. of Intl. Conf. on IEEE SMC'06*, 2006.

[10] D. Xie, T. Yan, D. Ganesan, and A. Hanson. Design and implementation of a dual-camera wireless sensor network for object retrieval. In *Prof. of 7th Intl. Conf. on IPSN '08*, pages 469–480, 2008.