A web-based cooperative tool for risk management with adaptive security

Mariagrazia Fugini [a], Mahsa Teimourikia [a,*], George Hadjichristofi [b]
[a] *Via Ponzio 34/5, Department of Electronics, Information, and Bioengineering (DEIB), Politecnico di Milano, Milan, Italy*
[b] *Department of Computer Science and Engineering, Frederick University, Nicosia/Limassol, Cyprus*

ABSTRACT

Risk management can benefit from Web-based tools fostering actions for treating risks in an environ-ment, while having several individuals collaborating to face the endeavors related to risks. During the intervention, the security rules in place to preserve resources from unauthorized access, might need to be modified on the fly, e.g., increasing the privileges of risk managers or letting rescue teams view the exact position of the victims. Modifications should respect the overall security policies and avoid security conflicts. This paper presents a dynamic access control model for environmental risks involving physical resources. Data structures included in our Web application to represent both risk and security are given. To keep the dynamic security rules compliant with overall organization security objectives, we consider rules grouped in *Access Control Domains* so that changes do not create security conflicts during collabora-tion in risk management. Considering work environments as an example, risk and access control models are introduced. Security is built on the ABAC (Attribute Based Access Control) paradigm. A Risk Man-agement System (*RMS*) is illustrated: it captures events, signals potential risks, and outputs strategies to prevent the risk. Dynamic authorization is included in the *RMS* to vary subjects' privileges on physical resources based on risk level, people position and so on. These concepts are implemented in a prototype Web application appearing as a Web Dashboard for risk management.

1. Introduction

In this work, we provide a framework for security within the context of risk management using smart data for cooperation in risk management in the Semantic Web. Networks, sensors and the underlying technology are great about moving information, as required, for instance, in environments where risk can arise. However, little is usually known about the data itself, namely about the semantics embedded in the captured events that signal a potential risk. Moving towards smart environments requires that data capturing risks and security is considered as an entity by itself that starts to live from the time of creation. As time progresses, this data item gets enriched with more information coming from modules performing the interactions with the entities such as other events, individuals, sensors, machinery, and so on.

Thanks to the availability of a large variety of sensors and devices that both sense and integrate data into technological monitoring platforms, several information can currently be gathered from the environment to monitor its status and possibly notify risks and critical events [1]. The environmental data items gathered in this way can be automatically processed [2] to signal, usu-

* Corresponding author.
E-mail address: MAHSA.TEIMOURIKIA@POLIMI.IT (M. Teimourikia).

Published Journal Article available at: http://dx.doi.org/10.1016/j.future.2015.04.015

ally in the form of events, the changes in the environment conditions (e.g., the temperature). A step ahead consists in interpreting these events to understand the risk that is possibly arising, to elaborate a strategy to prevent the risk, and to support the execution of the strategy by involved actors such as risk managers.

Since for efficient risk management information need to be visualized and inspected by algorithms and by risk managers to decide how to intervene, there is a need to perform data analysis from the perspective of risk teams. So they can be supported in understanding risks and intervening for risk treatment and minimizing the damages. Weitzner et al. [3] have motivated this direction and have demonstrated the concept through the popular Semantic Web cake developed for Web 3.0. An aspect we are dealing with in this paper is *dynamic access control* in risk management. In fact, in case of risk, security rules might need to change, for instance to increase clearances of risk managers, or to augment the capabilities of observing a scene by obtaining details on the people in the environment, so possibly overcoming the privacy rules applying normally. By taking into account issues related to cooperation of risk management teams and employing the Semantic Web for capturing the different nature of data items used for efficient risk management, we propose data structures to represent risk itself as a data item endowed with semantics in order to treat it through smart prevention strategies. We also present a risk-aware adaptive access control model allowing different actors to work on different views of the environment and to vary their privileges at run time.

The theme of cyber-security in facing environmental risks is becoming popular [4]. Risks considered here arise for example in plants or work areas, where the activities, the material, the tools or the people can cause harm or introduce threats to the health or life of the workers, or damage the resources and infrastructures inside the environment. As the observation on the environment progresses, data collected and constituting the risk-related events get enriched with more information received from sensors and devices. Also, the interactions with the world, such as changes of individual's activities and locations, changes of environment conditions such as temperature and humidity, and changes in the usage of machinery can modify and enrich the events. These data need to be inspected according to privileges of users but also in more details if an event signals an arising risk. Accordingly, in this paper, we aim at: (1) Proposing an access control model defining how security rules can be modified dynamically (e.g., allowing access privileges to be selectively shared within cooperating teams) while respecting the overall security policies to not violate data confidentiality and privacy; (2) Facilitating the collaboration between different actors that work with each other to treat the risk; (3) Proposing a method to efficiently manage risks by risk management team members based on the semantic nature of the data items and the security rules in place.

Environmental risk management embraces various themes, from technologies of surveillance to strategies of risk treatment and security and privacy of the involved resources (see [5]). Adaptivity and flexibility of access control models according to knowledge and data about the context are topics currently popular in various areas of data management, information systems and Web applications [6]. By using appropriate model concepts, individuals can be granted privileges, temporarily, in a selective manner on resources to prevent the risk once knowledge about risk has been assessed, and then return to the previous state of access control having risk-related privileges revoked. We present a risk-adaptive access control model, assuming a preventive approach (risks are handled before they turn into crisis). Our purpose is to have a system able to "reason" about risks by employing a probabilistic approach about events arising in the environment using data collected from the environment (see [7]).

We propose a method based on Event–Condition–Action (*ECA*) meta-rules which allow modifying the security rules according to the risk characteristics that we consider as attributes and based on the preventive strategies that should be executed. We describe a *Risk Management System* (*RMS*) (see [7]) able to elaborate on events and risks and to cooperate with an *Access Control System* (*ACS*) that is in charge of authorizations based on security rules. Such cooperation also considers the concepts of smart secure data in the Semantic Web and sets the basis for a framework that allows risk and security-related data to evolve and capture the aspects of security and risk management in dynamic environments. In this effort, we focus on access control through the Attribute-Based Access Control (*ABAC*) paradigm [8]. Since Web-based collaboration can be useful during risk management, by providing a unified dashboard supporting the sensing of risk notifications and risk treatment strategies to the risk teams, our *RMS*, which elaborates on events and outputs the description of risks and the suggested strategies, supports cooperation via the Web to help:

1. Recognizing the risks and their attributes from the monitored environment.
2. Cooperating with the *ACS* providing information to be used to adapt the access control rules to treat the risks efficiently and effectively.
3. Supporting risk management by actors in a cooperative way, relying on the risk strategies and communicating with the *ACS* for access control adaptation.
4. Representing environment, risk, security rules, intervention strategies, and individuals as an entity, the *RMS* incorporates knowledge about the process of risk management (which is a business process of activities).

The paper is organized as follows. In Section 2, we set the overall data structures to be shared on the Web to generate risk and security knowledge. These have then to be adapted to each single environment. We also give definitions of risk and security and present our approach to computation of risk and to cooperation in risk management. In Section 3, we illustrate the model of risk-adaptive security, and give the detailed concepts needed to handle adaptivity in security rules. In Section 4, we discuss the *RMS* architecture and the security rules implementation in XACML to obtain a cooperative Web Dashboard for risk and security awareness. Then we present a scenario (work areas) where use cases related to the risk and security concepts are exemplified. In Section 5, we review the relevant state-of-the-art articles and systems. Finally, in Section 6, we draw conclusions and outline future work.

2. Framing data structures

In this section, we set a framework of data that constitute the basis for elaborating on adaptive security in risk management actions. In particular, we state some preliminary definitions of risk and security needed to form a collaboration system for risk mitigation, which is expected to rely on several security aspects [9], such as *accountability* or *authentication* and *access control* mechanisms [10]. For systems in which unknown users may enter, the concept of trust is usually used to monitor users behavior along time [11].

In this work, we deal with *security rules for access control*, while accountability, authentication, and trust are not investigated. We assume these issues are treated outside the *RMS* and the *ACS*, and that people in the environment are known and have been authenticated. Leveraging our previous work on risk [12], we refer to a sample scenario given by risk and security in a *work environment*. Here, Workers (Subjects) use tools and machinery (Objects) in a set of work areas (Environment sections). Subjects need to have specific Privileges on Objects to operate in the Environment. These Subjects are potentially exposed to risks. Risk is treated using a probabilistic approach, considering that it can

be detected and handled before it grows into a crisis. Risk is modeled as an entity connected to Subjects and Objects operating in the Environment as shown in Fig. 1. It is a data structure semantically annotated using attributes describing the risk type, its severity, and other factors which contribute to determine the risk itself (monitoring tools present in the environment, protection devices, and so on). Subjects are also actors responsible for risk management, such as risk teams and members, departmental units, rescue groups, and so on.

Coming to security, an important issue is the possibility of *fine-grained adaptation* of access controls to risks. In case of a risk (e.g., fire), adaptive accesses modify the capabilities of Subjects to view the system resources (Objects) for risk management: access rules can vary at run time to handle the risk and be later revoked when the risk has been managed.

Finally, it is important to have security Privileges that apply to physical resources in the environment, (e.g., "open a door", "activate sensors") and that represent also procedures whose execution must be authorized, as defined by Paja et al. [13].

In what follows, we elaborate on the Risk data structure linking it to other model concepts.

2.1. Data structures

In this section, we define the data structures linking risk and security. We refer to ISO 31000, as a standard for Risk Management [14] and the source of concepts related to risk. Fig. 1 represents the data structures for Risk. The Subject and Object in Fig. 1 will be detailed in Section 3. Relationships represent aggregations, compositions and Is-A hierarchies of concepts.

(1) *Environment*. As represented in Fig. 1, the *Environment* (*EN*) is a physical area, both open or closed, monitored for events. Different parts of *EN* can be equipped with various sensors and monitoring devices (*MD*); the risk can affect only some parts of *EN* and therefore different risk treatment strategies may take place in different parts. For this reason, we consider *EN* to be composed of a set of *sections* en_i that can be monitored for risks. Specifically, $EN = \{en_1, en_2, \ldots, en_l\}$, in which $l \in \mathbb{N}$, $l \leq MaxSections(EN)$, and $MaxSections(EN)$ returns the total number of sections in a given *EN* that is usually extracted from the buildings blueprint or a map. The *Environment* sections are where people can congregate, such as rooms, areas, corridors, doorways, and etc. *EN* contains many elements different in nature and type, such as people (*Subjects*), tools/machinery, devices, materials, and so on. These are all considered as Objects to be protected. In addition, we consider that *EN* sections are linked to one another or to the outside world via *Connections* (*Conns*), such as doors/windows. *Conns* are useful to establish escape plans and exit ways and paths.

(2) *Event*. As depicted in Fig. 1, we use the concept of event to represent a change in the environment parameters that are monitored, such as temperature, CO_2 concentration, or based on the Subject's monitored conditions or activities, like working with a hammer, or a very high heart beat. In other words, events can be considered as incidents or accidents that have causes and consequences and are raised by a parameter exceeding its legal values. The cause of an event is referred to as the *Event Source*, and the con-sequence of the event is referred to as *Risk*. The Events might have no consequences, and be called *near-misses*. Events can have one or more Event Sources and one or more Risks. Examples of events in an industrial work environment are:

- Gas leak;
- Fuel leak in one of the fuel farms;
- Inappropriately-equipped personnel working with a tool with high risk of injury;
- A detected abnormality in the health conditions of monitored staff.

(3) *Risk*. Risks are the consequences of Events. According to ISO 31000, the risk is an occurrence that might affect the organizations and have different consequences on different areas, such as economic performance or the reputation of the organization [14]. In our approach, we consider *environmental risk*, where a risk $r_i \in R$ (being *R* the set of all risks) is a hazardous situation in the environment that can cause harm to people, resources, and/or structures in the monitored locations. To set an example, if the temperature in a given area is higher than the defined normal threshold, the temperature parameter is signaled as an event and might generate a risk. We assume that parameters are evaluated using a predefined approach, such as a threshold-based approach or a range-based ap-proach, depending on the parameter at hand [7]. We model *global risk* and *individual risks* as shown in Fig. 1. The former are risks af-fecting parts or the whole environment. The latter affect a person (which can further cause a global risk). The reason for this distinc-tion is the different methods used to treat risks. The recognition of the individual risks can be very fine-grained (e.g., checking if an individual is endowed with all the necessary safety garments when entering areas associated with risks of injury). When facing a global risk, a fine-grained recognition and treatment of risk for each individual requires more complex methods and is hard to obtain, because of the time sensitivity, of the unavailability of moni-toring and tracking data on each individual, and of the complexity of such approaches combined together. Conversely, in these cases, we consider a general recognition of the risk for the affected people, areas, and physical resources, and determine the strategies to face the risk (e.g., closing the gas flow in a section of the environ-ment where the temperature is higher than a safe threshold and is at risk of fire). In order to detect and manage the risk, we consider the following model entities:

- *Risk source*. If an Event consequence is a risk, then the Event Source is considered as the Risk Source. The *Risk Source (RiskS)* is the physical element causing the risk (e.g., a source of fire). It is relevant since the closeness of people to the source gives the values necessary for our system to evaluate the risk level. If there are various sources causing a risk, we should consider the combined risk deriving from them. E.g., water flooding that occurs close to electric plugs gives a higher risk level, and this level is not just the sum of the two individual risks. In the probabilistic risk model, we do not consider mutual-dependent risks but risks occurring one at a time [7].
- *Monitoring devices*. Monitoring Devices (*MD*) are the technological elements (RFIDs, sensors, wearable devices [15], cameras, etc.) that are in place in the environment or carried by people or attached to resources to monitor the Risk Sources. *MD* forward information to the *RMS* and are used to notify events that may cause risks.
- *Informative devices*. Informative Devices (*InfD*) are the technological elements used to acknowledge persons in the environment about risks, strategies, and alarms, such as: PDAs, smart phones, tablets, environment alarms, etc. We consider that persons endowed with tools (smart phones, communication means) and participating in risk management are cooperating actors that need to collaborate during all the phases of risk management, from its detection to its conclusion.
- *Protection elements*. Protection Elements (*PE*) are devices and tools used for risk prevention and safety that provide protection for people, resources, and the environment. These devices can be wearable sensors or elements which are in place in the environment for safety reasons, such as gloves, visibility vests, protection glasses, fire extinguisher or helmets.

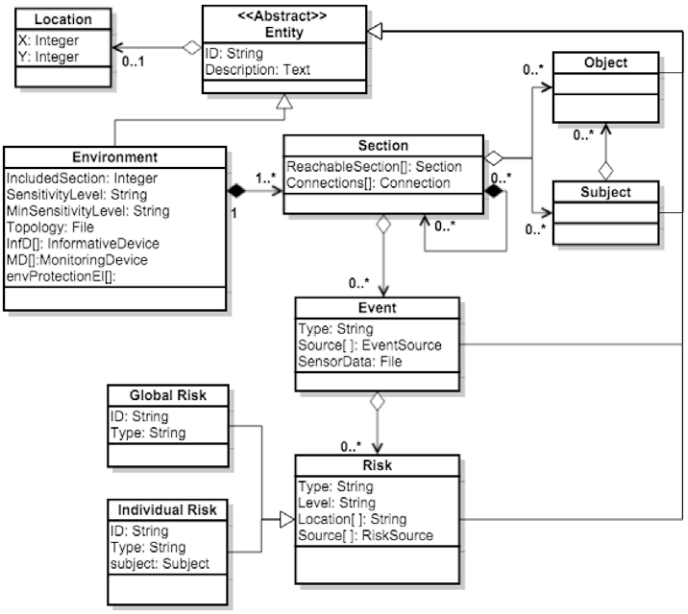

Fig. 1. Data structure for risk.

- *Tools and machinery*. People use tools (e.g., hammers, drills) or machinery (e.g., trucks) to perform a task or to move in the environment. In using these tools, a risk can arise which we model as a link between a Subject and an Object during a work activity.

2.2. An introduction to risk computation

Now, we revise our approach to risk management constituting the basis of the *Risk Management System* (*RMS*) presented in detail in [7].

The *RMS* is a Web application whose front-end is a portal for risk-related services and risk teams coordination and cooperation. Its back-end incorporates the procedures for event, risk, and strategies management embedded in an application logic aimed at risk prevention. The data layer stores the data structures defined for risk and security. The outcomes of the application logic are: a set of actions (strategy) to be undertaken to mitigate the risk, and a set of security rules that need be granted temporarily to face the risk and later revoked. Considering *EN*, in case an event *e* is signaled by *MD*, the *RMS* analyzes *e* to recognize the risk *r* associated with *e* and to suggest preventive strategies for treating *r* before it evolves into a crisis. This is done during a set of *risk management phases* which follow the MAPE (monitoring, analyzing, planning, and executing) pattern [16]:

- Monitoring the environment parameters;
- Analyzing the data about monitored values, recognizing the out-of-range parameters, and extracting the event *e*;
- Planning preventive strategies based on the results of the computation of the risk *r* according to the probabilistic model of risk presented in [17]. The output is a description of *r* which includes all the risk attributes that will be described in the following of the paper and constitute the Risk entity; the strategies to prevent the recognized risk *r* are computed;
- Executing the automated strategies and supporting the risk team to execute the preventive strategies assigned to them. A

Gaussian distribution for each risk source [17] allows modeling the maximal value in the origin of danger that has an inverse relationship with distance, meaning that it decreases when the distance increases and vice versa. The distribution of values around the means denotes how flat the curve is, namely, to which extent the danger continues to influence the risk value while the distance increases. We consider the distribution of risk in a bi-dimensional area (X, Y), on which we draw a tri-dimensional diagram where the Z axis represents $r \in R$ connected to e, fully computed in its characterizing attributes: *ID* that uniquely identifies the risk; *RiskLevel*, denoting the risk intensity that we express using a finite set of values, namely VERY HIGH, HIGH, MEDIUM, LOW, and VERY LOW; *RiskType*, featuring the type of risk (e.g., risk of fire, of fall, etc.); the *Risk Source* that caused such risk (e.g., a truck, an electric wiring, etc.); the *locations* affected by the risk that are identified by the en_i. ID (*Location*); and the strategies proposed by the *RMS* to treat *r* (*Strategy*).

The computation of risk is a continuous procedure composed of the steps illustrated in the following.

The first step is monitoring and data collection, where data are collected by *MD* as data streams [7]. The basic *monitoring functions* are: *Signal Detection*, in which *MD* produce data streams describing the monitored entity; *Localization*, that allows tracking and monitoring the 2D coordinates of workers and of moving objects (e.g., trucks); *Tool Monitoring*, that, using passive RFID tags or other sensors attached to tools, allows continuously monitoring the tools/machinery carried/driven by workers; and *Reporting*, that generates logs of the variations of monitored parameters (which can possibly cause risks).

After data collection, the *RMS* analyzes data for risk identification. Here, the collected data are evaluated and correlated to the concept of risk in the environment using the *Risk Evaluation Function (REF)* defined as follows:

$$EvaluationValue = REF(parameter) \quad (1)$$

where *REF* denotes an evaluation function and *parameter* represents the monitored elements used to compute the risk. The *EvaluationValue* is a normalized numeric value $\in [0, 1]$ expressing the risk associated to each parameter. Each *parameter* is associated with a particular threshold for that *parameter*, and $EvaluationValue[parameter] > threshold$ indicates a potential risk. A set of *REFs* are stored in the *RMS*, associated to the various monitored elements, to feed the *RMS* algorithms for risk computation and strategy elaboration.

The *RMS* is *user-centered*, namely, aimed at preserving people's safety, as a first purpose, and at giving risk management teams

instructions to collaborate and intervene. Therefore, beside risk-related elements, such as risk type, intensity, risk location, and risk source, which determine the global risk, the *RMS* calculates two other values: the individual's *Personal Risk Level (PRL)* and the *Personal Protection Level (PPL)* to determine the individual risks. Let us assume that the *RMS* computes a set of risks R; for simplicity of illustration, we consider one risk $r_i \in R$ at a time. This is continuously computed both for individuals and as a global risk for *EN*, that is updated by the *RMS* (see [7]).

2.3. Cooperation in risk management

It is evident from the definitions above that different types of risks differ significantly in functionality and typology. Moreover, the actions taken by individuals in facing the same risk may differ widely. Nevertheless, risk should be handled cooperatively under a unique yet distributed architecture and key components. The *RMS* is in charge of managing the overall strategies for risk treatment. In particular, actors (subjects) can cooperate:

- directly:
  - exchanging alarms and messages about what happens in the environment;
- indirectly:
  - in a centralized way via the *RMS* which then elaborates a strategy and communicates it to all the actors who are required to intervene;
  - in a distributed way via the *ECA* module, which evaluates the Events, Conditions and Actions for subjects requesting to access resources. To access resources, subjects must be assigned with roles associated to a set of authorizations. These roles can vary at run time due to the risk mitigation needs and are processed by the *ACS* module.

Considering indirect cooperation, which is interesting for our case, centralized cooperation through the *RMS* needs shared data structures describing Monitoring Devices *MD* (e.g., a sensor or a surveillance camera) and a set of functional modules of the *RMS*, each embedding a device controller and a *RMS* message processor. In fact, we consider that various devices and tools have an interface to cooperate with the *RMS*; actors can cooperate via the *RMS* in a peer-to-peer way also to assign roles and privileges to people. The overall coordination occurs passing through the *RMS* which collects the data from devices and, based on its internal knowledge about events and actors, outputs a strategy of intervention. In the second case, namely cooperation in a distributed way via the *ECA* module, we have a further step that needs evaluation of local rules in place at the various system nodes and an evaluation of dynamic authorizations. We assume actors communicating via predefined communication channels to exchange messages in voice or text or other media formats and a decision taken in a cooperative way on the basis of the *ECA* and *ACS* modules decisions. To obtain a highly coordinated and resource-constrained system, a collaboration agreement between two *cooperating actors* can be represented by means of a collaboration relationship. Collaboration relationships in our approach are oriented to define the level of security and privacy that should be ensured between actors and is managed by the meta-rules designed as *ECA* rules to decide how to adapt the *ACS* to allow the cooperation of actors to treat the risks and to follow the strategies suggested by the *RMS*. Our solution provides a model where actors can describe which specific information from their information models is shared with other actors. If an actor a collaborates with another actor b, this implies that b has available certain risk-related information of a to describe its security rules. Thus, a collaboration relationship can be seen as a triple (*Grantor-Actor*, *Grantee-Actor*, *ContextData*) describing that a Grantor manages the authorizations of a Grantee by exposing the context information to him and hence can modify the authorizations of the Grantee temporarily to handle the risk. Using context information, we consider that users (Subjects) can administrate their privileges so that risks can be handled cooperatively. The Grantee can use information belonging to the Grantor and use new authorization statements. This mechanism enables cooperating actors to control the access to the resources from each other. A common situation is an actor b allowing another actor c to use its resources – knowledge, data, tools, etc. – (indicating these in the data structure named ContextData). It would permit c to operate according to security rules taking into account resources of b. This approach enables b to grant his resources temporarily to c. It could also happen that the collaboration relationship $a - b$ implies that b is allowed to use the resources defined in the security rules of a under conditions constraining the access (e.g., time constraints). To do so, a only needs to specify that information within the ContextData is shared in the collaboration relationship. As an example of this situation, an actor a could describe an authorization rule stating that a user who is in turn in a collaboration relationship with actor b could access some of the a's resources only if b's smart phone communications of a given section of the environment are not available. It is important to remark that this feature does not imply that a can grant privileges for his users to access b's resources. This can only be specified by b, namely: an actor is the only one who can determine its permissions over its resources and authorizations cannot be transmitted with the grant option (the grant chain is not allowed). Collaboration relationships are not transitive so that we do not need automatically manage such issue in order to keep the federation model simple. Obviously, this is supported just by explicitly inserting such collaboration relationships. Moreover, collaboration relationships are not symmetric. This enables more control over the definition of federated security scenarios. Analogously, in case this feature is needed, the inclusion of the corresponding symmetric collaboration relationship is enough to support it into the system. Finally, by default nobody is in a collaboration relationship with anyone else unless there is an explicit statement for this. The collaboration relationship information is used by the *ACS* module available in the architecture. Upon a request, this module is in charge of providing only the knowledge of those organizations which collaborate to each other to take the authorization decision. It controls the privacy of all the information stored in the system database(s). The module selects the information to be used to make the authorization decision taking into account the issuer requesting authorization in order to provide multi-actor risk management in the environment.

Our architecture assumes a different database of risk knowledge is kept for each actor. Each contains the information model and the security statements for a particular actor. Each database also keeps the knowledge allowed by its cooperating actors according to the collaboration relationship model defined with the aforementioned triple. This approach is safer and more secure than having just one database for all the information, since it allows us to physically isolate sensitive information from different actors and also in case of unavailability of some portions of the system. The *ACS* also manages the life cycle of the different databases keeping them up-to-date according to the changes in the collaboration relationships between actors. This update implies removing knowledge from the database when an actor cancels a collaboration relationship or updating the databases when a new relationship is established. Holding different databases with information for different actors implies that the *RMS* also has to be aware of these relationships in a historical way so as to update its strategies according to the information coming from the occurred events.

3. Risk-adaptive access control model

Here the data structures used for the risk-adaptive access control model based on ABAC are defined. We refer to the module in charge of the access control as *Access Control System* (*ACS*).

Subject models users and entities that take actions in the environment, while *Object* models resources to be protected and requiring authorization to be acted upon. Objects are physical resources, such as areas, tools, or anything that needs authorization to be operated upon.

Definition 1. *EN*, *S*, *O*, *P*, *R*, *RU* and *ACD* constitute the finite set of *Environment Elements* and the existing *Subjects* and *Objects*, *Subject's Permissions* over *Objects*, *Risks*, *Security Rules* and *Access Control Domains* respectively.

Definition 2. *ENA*, *SA*, and *OA*, denote a finite set of *Environment*, *Subject* and *Object* Attributes. An Attribute is a function defined on an *Environment*, a *Subject* and an *Object*, which returns a specific value from its range. Attribute values are either atomic, namely a single value in the specified range, or are given as a finite set.

Definition 3. *Access Control Domains*, *ACD* are sets of security rules (*RU*) that apply when given groups of risk attributes are recognized in the environment. *ACD* allow us to enforce the need-to-know policies and the confinement properties of security [18].

In what follows, we introduce the data structures used in the access control model.

3.1. The subjects

Subject $s \in S$ is an entity that abstracts a person, a process or any "active entity" in the environment needing permissions to act on resources. Subjects are considered in three different categories:

Administrative subjects: These are a group of authoritative people, who collaborate and balance one another decision/control. Their main responsibility is to assign the Subject, Object, and Environment Attributes.

In-domain subjects: These are the users inside the organization identified and authenticated by the *ACS*. In our scenario, they are the workers including the staff and personnel in organizational divisions like: surveillance, security, maintenance, ground transportation, production, customer service, risk management and safety, human resources and so on. These Subjects are active users that need permissions to access different resources. Also, they need to be protected from the risks that might affect them.

Out-domain subjects: These subjects are from outside the organization, including the visitors inside the different areas of the environment. We consider them as passive Subjects who do not need to access the resources in the environment. However, in case of detection of a risk, they need to be protected. A few information might be available on these Subjects. For example, their complete identity, their qualifications, scope of presence in the environment and their exact position might not always be fully available. However, using monitoring devices, it is possible to understand which sections of the environment are, for instance, most crowded, so prioritizing the strategies that allow rescuing the largest number of people [7]. These Subjects might be unidentified Subjects, or identified but not authenticated Subjects. As soon as an out-domain Subject is identified and authenticated by the *ACS*, he will be considered as *In-domain Subject*.

Subjects are modeled by a class diagram as shown in Fig. 2.

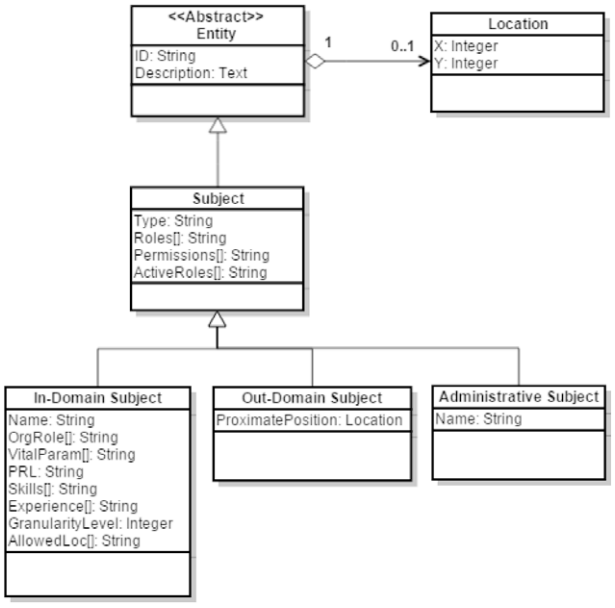

Fig. 2. The subject data structure.

Fig. 2 depicts the *Subject* data structure in our access control model. In the following the attributes of an In-domain Subject are shown:

$s_i \in S$: {**GeneralAttributes** : {*ID*, *Name*, *OrganizationalRole* {*Skills*}, {*Experience*}},
RiskAttributes :{{*ProtectionElement*}, {*VitalParameters*}, *PRL*, {*MonitoringDevice*}},
GeoAttributes :{*Location*, *GranularityLevel*},
SecurityAttributes :{{*Roles*}, {*ActiveRoles*}, {*Permissions*}}}.

GeneralAttributes define the general characteristics of Subjects, such as their *ID*, *Name*, and *OrganizationalRole* (*OrgRole*). The *Type* attribute indicates the aforementioned category of Subjects. For In-domain Subjects, we also consider *Skill* and *Experience* sets that indicate the ability to use tools and machinery and their knowledge of security procedures and work organization that is related to length of involvement in working with each aspect. These are expressed using an index (low, medium, high, very high) for each tool, machinery and aspect.

RiskAttributes for In-Domain Subjects include: the list of *PersonProtectionElements—PPE* that the Subject is endowed with, such as hard helmets, safety glasses, high visibility vests, and etc.; the list of *VitalParameters* (*VPar*), like the heart beat or the body temperature received from the *MD* carried by the Subject (e.g., wearable sensors); the Personal Risk Level (*PRL*) that is computed by the *RMS* [7] on the basis of risk exposure, protection elements, position of the person wrt Risk Source, and so on.

We also consider **GeoAttributes** for the Subjects. Specifically, for In-Domain Subjects, that can be located precisely since they are endowed with wearable sensors and devices that facilitate the localization process, we consider the *Location* (*Loc*), and the *GranularityLevel* (*GranL*), which denotes the level of details for which the Subject's position data are available for privacy reasons. For example, the exact location of the Subject might be hidden while the Subject's logical position is available.

Furthermore, **SecurityAttributes** include the list of *Roles* that the Subject can have, the list of *ActiveRoles* at each moment, and the list of authorized *Permissions*.

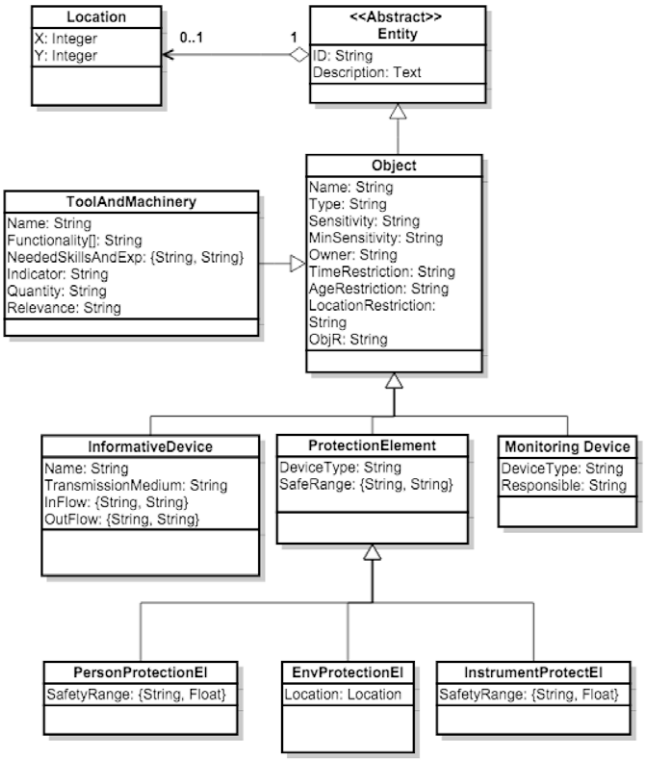

Fig. 3. The object data structure.

3.2. The objects

Object $o \in O$ abstracts resources that a Subject can access or act on. We consider physical resources as Objects (e.g., doors, gas pipes, electricity panels, tools, machinery) in that they can be operated upon.

We assume four categories of attributes for Objects, namely General, Risk, Geo, and Security Attributes:

$o_i \in O$: {**GeneralAttributes** : {*ID*, *Type*},
RiskAttributes : {{*ObjR*}},
GeoAttributes : {*Location*, *GranularityLevel*},
SecurityAttributes : {*SensitivityLevel*,
MinSensitivityLevel, {*Owner*}, {*TimeRestriction*},
{*AgeRestriction*}, {*LocationRestriction*}}}.

GeneralAttributes include the unique *ID* of the resource and its *Type*.

GeoAttributes correspond to the *Location (Loc)* of the Objects; if the Object is moving, the location attributes will change as the Object moves. If the Object is fixed, like sensors that are installed in a specific location, a fixed attribute will represent the Objects location. The *GranularityLevel (GLevel)* shows the level of precision in identifying the Object's location. The *GranularityLevel* can be *precise*, which gives the exact coordinates of the Object, if applicable, or *logical* that can roughly show whereabouts of an Object, e.g., if the *GranularityLevel* is set to *Section* then it is possible to know in which section the object is.

RiskAttributes include the *ObjR* indicating the level of risk a single Object is exposed to, or introduced in the environment (if for instance it is a work machinery moving in a section where people work). This value can be one of the following: VERY HIGH, HIGH, MEDIUM, LOW, VERY LOW. For example a hammer *ObjR* value is set to HIGH as it can introduce a risk of injury to the Subject using it.

AccessControlAttributes define security-related attributes that show how the Object can be accessed by different Subjects. These attributes include: current *SensitivityLevel (SL)*; *MinimumSensitivityLevel (MSL)* that an Object can get at run-time; the Object's *Owner*; *TimeRestriction (TimeR)* that sets a time limit in which the Object can be accessed; *Age/LocationRestrictions (AgeR/LocR)* that limits the age or the locations of the Subject who requests a permission on the Object. The *SensitivityLevel (SL)* of an Object defines the level of security needed to access that Object. These levels are defined as SL_i in which, $i \in N$, is an index where the higher the index the more sensitive the Object. Since attributes might change at run-time, in order to avoid highly sensitive data to be exposed to downgrading their security level, we consider a *MinimumSensitivityLevel* attribute for Objects.

Fig. 3 represents the Object Entity.

Physical Objects are the resources that can be accessed or act upon physically. The method of access to these Objects is indicated by the *AccessType* attribute (e.g., Passing Swipe Card, Entering Access Code, etc.). In our scenario, we consider the following samples of the Physical Objects:

Tools and machinery, Attributes of *Tools And Machinery* include: *ID* to uniquely identify them, *Name*, *NeededSkillsAndExperience* denoting the skills and experience of the Subject that ensures their proper use, the *Usage Instruction* of the Tool/Machinery, including the *Indicator* (e.g., usage time, distance, etc.) and *Quantity* that limits the allowed variations of the *Indicator* (e.g., (2–3) Work Hours, $\leq n$ meters). The *Relevance* attribute indicates the risk level if the current usage instruction is not followed (e.g., VERY HIGH, HIGH, MEDIUM, LOW, VERY LOW).

Informative devices and monitoring devices, include attributes such as *ID*, *Name*, *Transmission Medium*, specifying the communication medium between the device and the *RMS* (e.g., wired connection, Wi-Fi, Bluetooth), *InputDataFlow* that shows the *Channel* and the *DataType* used for receiving data from the *RMS*, and *OutputDataFlow* that shows the *Channel* and the *DataType* used for sending data to persons (for Informative Devices) or to the *RMS* (for Monitoring Devices).

Protection elements (*PE*), Three categories of PE are considered:

PersonProtectionElements (*PPE*), e.g., garments, providing physical protection for Subjects (e.g. helmet, safety glasses, high visibility clothing, foot protection, etc.). Protection Elements are characterized by the *SafetyRange* attribute, denoting the range of parameter values in which the garment will guarantee the safety of the Subject.

EnvironmentProtectionElement (*EPE*), refer to the elements present in *EN* that can be used against the risks (e.g., sandblasting equipment, or fire extinguisher). This element has a *Location* attribute that shows the geocoordinate where it can be found.

InstrumentProtectionElement (*IPE*), provide physical protection for devices, tools and machineries, such as handling gloves, masking shelters, emergency brakes, emergency doors. They are also characterized by *SafetyRange* attribute, denoting the range of parameter values wherein the protection element will guarantee the safety of the instrument. Their position is equal to the position of the protected tool: if the protected tool is moving, the localization is dynamically modified accordingly.

3.3. The environment

An environment en_i is partitioned into sections to simplify its surveillance and management and to reflect the organization of environments as they are in the real world as described by maps, blueprints and so on. Each $en_i \in EN$, $1 < i < n$ has some attributes

$ena_i \in ENA$ that are themselves finite sets. These attributes are considered in four categories, namely, General, Risk, Geo, and Security Attributes:

$en_i \in EN$: {{**GeneralAttributes** : {*ID*, *Description*}},
{**RiskAttributes** : {{*RiskSources*},
{*MonitoringDevices*}, {*InformativeDevices*},
{*ProtectionElements*}}},
{**GeoAttributes** : {*Location*, *Topology*, {*Connections*},
{*ReachableSections*}, {*IncludedArea*}}},
{**SecurityAttributes** :{*SensitivityLevel*,
MinSensitivityLevel}}}.

GeneralAttributes including *ID* and *Description*, refer to the group of attributes describing the general characteristics of en_i.

RiskAttributes characterize the risk-related aspects of en_i, namely, *RiskSources* that are the elements existing in en_i which can potentially create risk events (e.g., electric plugs, gas emission points, etc.). *RiskSources* also include the Tools and Machinery in en_i that can be potential risk sources with which Subjects can interact (e.g., power drill, baggage trailer and truck refueler). The *RiskSources* are themselves Objects; hence they are referenced in the list of Object *ID*s. *ProtectionElements (PE)* refer to the list of Object *ID*s present in en_i that can be used against the risks (e.g., sandblasting equipment, and fire extinguisher). *MD* are the list of Object *ID*s that are used to monitor parameters and entities of the section (e.g., sensors, video cameras, x-ray and screening machines). And *InfD* are Objects present in the section that can be used to notify the risk events (e.g., alarms).

The third category, **GeoAttributes**, contains attributes such as the *Location (Loc)* of the section that is shown as the Cartesian coordinates of the right-up corner of en_i, the geo-spatial *Topology (Tp)* of the section that is defined in a file, its *Connections (Conn)* that are the list of *ID*s of in and out openings like doors and windows that allow entering and exiting the section, and a list of *ID*s of the first *ReachableSections (RSec)* from en_i. *Connections* include the following attributes: *ConnectionType (CType)* denoting if the connection is for entering (In), exiting (Out) or both (In/Out). And the *EmergencyExit (Emg)* assuming a Boolean value that shows if the connection is an emergency exit. Also, *IncludedAreas (IA)* show the number of sub-sections inside the described area.

Finally, **SecurityAttributes** include the attributes used to indicate the *SensitivityLevel (SL)* of the section, which is denoted by $SL_i, i \in N$, where the lower the index i, the less sensitive the section as far as security requirements. *MinSensitivityLevel (MSL)* marks the minimum sensitivity level that can be assigned to a section dynamically while manipulating the attributes.

3.4. Privileges

A privilege $p \in P$ is the operation that a Subject requests to perform on an Object. Privileges are either actions on data, such as read, write, execute, create/delete, or actions such as: trigger (for alarms), turn on (for electricity power switch), save (for a rescue team on people at risk), or complex operations described as activities or tasks, with their internal description as a process.

3.5. Adaptiveness of the access control model to risk

To add dynamicity and adaptiveness to the *ACS*, we consider changing or setting entity attributes dynamically according to the risks that are recognized by the *RMS*. As defined in the previous section, the attributes of the access control entities can be either static or dynamic. Static attributes are set and managed by Administrator Subjects. However, the dynamic attributes such as location, risk related attributes, and security related attributes might change dynamically to have a dynamic access control model. Changes in dynamic entity attributes are managed through the *ECA* paradigm [19] to define meta-rules that set the conditions for dynamic attributes to change.

Context are represented using *ACD*, namely pools of resources and security rules available in a given situation, specifically activated/deactivated depending on what occurs in the environment. *ACD* data are *semantic structures* stating which Subjects can operate on which Objects and can grant to other Subjects privileges in order to allow for cooperation.

As a sample set of *ACD* in a work area, we have the Risk Context (a possible explosion in the building area can be handled only by a specific team); the Open-Air Context (only sections situated open air can be inspected by the risk team personnel); the Office Context (engineers can operate in certain areas reserved for office activities); the Logistics Context (operations accessible for materials management by cargo units and staff).

A preliminary definition of Context is: a context C indicates a set of security rules to be valid in a certain situation based on dynamic changes in the environment. C defines the operations available to subjects on objects in an environment under certain circumstances. C contains also data of pertinence of the situation, so that data coming from sensors and devices can be clustered by contexts, and can be managed more efficiently (see [20] for details).

When a risk is present and several actors are cooperating to treat the risk, it is possible for the actors to require more privileges temporarily to be able to execute the strategies suggested by the *RMS*. These kinds of requests are forwarded to the *ECA* portion of the *RMS* where meta-rules are in place. These consider conditions in which the attributes of the entities, such as the Subject roles, can be modified to authorize the required actions. If there are no meta-rules that allow the request, then there will be no change in the entity attributes.

Later, when the risk has been treated, all the changes made to the entity attributes by the *ECA* module will be rolled back to the initial values. The *ECA* module is explained in more details in Section 4 and examples are given in Section 4.1.

4. Implementation and prototype

In this section we illustrate the architecture of the implemented risk-adaptive access control model. Fig. 4 depicts three modules, namely, the *RMS*, *ECA*, and *ACS*. *RMS* monitors *EN* and updates a database with detected risk attributes and strategies that have been evaluated by the module. The attributes of the entities namely *ENA*, *SA*, and *OA* in addition to the $RU \in ACD$ are kept in a database and are used by the ACS for authorization decisions. As shown in Fig. 4 a $Request \in REQ$ is submitted for evaluation to the *ACS* if a Subject needs Permission on an Object. A request is defined as the result of the application of an *evaluate* function as follows: $evaluate(s : S, p : P, o : O, en : EN)$.

Such request can be evaluated to "Permit" (Y), "Deny" (N) and "Not-Applicable" (NA) where $EFF = \{Y, N, NA\}$ is the domain of effects which is enforced by the *ACS* as depicted in Fig. 4. For instance, some privileges do not apply to the considered Object, such as 'Execute' on a 'Section Map' Object is undefined. The request is evaluated by the *ACS* considering the attributes of the Subject, Object, and the Environment, and the requested privilege and according to the applicable *RU* inside the *ACD*. Examples of an *acd* and its *rules (RU)* are given in Section 4.1.

Now, referring to Fig. 4, we describe the elements and modules able to dynamically adapt the *ACS* to different contexts. We assume that *ACD* are designed statically, while having the conditions that contain attributes updated in real-time, and therefore, the applicable *ACD* is chosen based on the dynamically-set attributes.

Table 1
Example of an environment entity.

ID	Tp	Loc	IA	RSec	Conn	MD	RiskS	InfD	PE	SL	MSL
Overall	TFileOv	(0, 0)	2	–	C0	–	–	–	–	–	–
SecA	TFileA	(0, 300)	0	SecB	C1	Surveil1	PowP1	–	–	SL_4	SL_2
SecB	TFileB	(400, 300)	0	SecA	C1	GasS1	GasP1	Alarm1	PE1	SL_3	SL_1

Table 2
Examples of subjects' attributes. Legenda: PPE = Personal Protection Element, PE = Protection Element, NA = Not Applicable; HB1 = Heart Beating; GranL = Granularity Level.

General attributes					Risk attributes					Geoattributes	
ID	Name	OrgRole	Skill	Experience	PPE	VPar	PRL	MD	InfD	Loc	GranL
W1	Mary Brown	Civil Engineer	Constructor Designer Digger	High Med Med	PE3	NA	Med	NA	NA	SecB	Section
W2	Paul May	Electrician	Electric eng.	High	NA	70 bpm	High	HB1	PDA1	SecA	Section

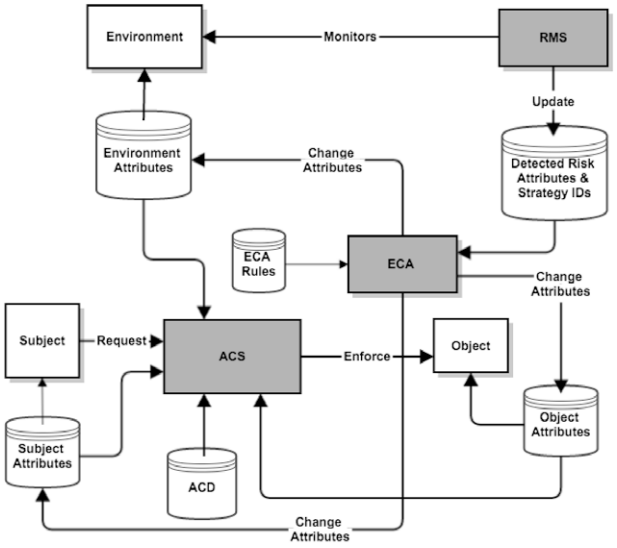

Fig. 4. An overview of the risk-adaptive access control.

Defining $RU \in ACD$ statically helps the overall set of rules to remain compliant with the security policies. Also, it is possible to evaluate the policies for inconsistencies and conflicts offline.

The *RMS* monitors the Subject, Object and Environment entities, and updates a database containing the detected Risk attributes and their related strategies. As shown in Fig. 4, the *ECA* module uses this database, and according to the meta-rules, changes the attributes of the entities.

The *ECA* meta-rules are implemented in the following format (see [19]):

$\langle meta\text{-}rule_{ID} \rangle$: **WHEN** $\langle$*Risk that causes the meta-rule apply*$\rangle$

IF $\langle$*Conditions*$\rangle$

THEN $\langle$*Actions*(*Changes in Entity Attributes*)$\rangle$.

A meta-rule is designed to govern the adaptiveness of the *ACS* by changing the attributes of the entities based on the risk events triggering it and the conditions that are set in the meta-rule. The *ECA* meta-rule contains three parts, namely: "*When*" indicates the risk event that triggers the meta rule; "*If* " contains the conditions; and "*Then*" includes the actions that mainly are designed to change the attributes of the entities.

Examples of ECA rules are given in Section 4.1. The *ACS* is implemented based on XACML's implementation as Balana [21]. XACML 3.0 [22] is the standard access control policy language that supports the ABAC paradigm. We employ XACML as the basis for implementation of our risk-aware access control model.

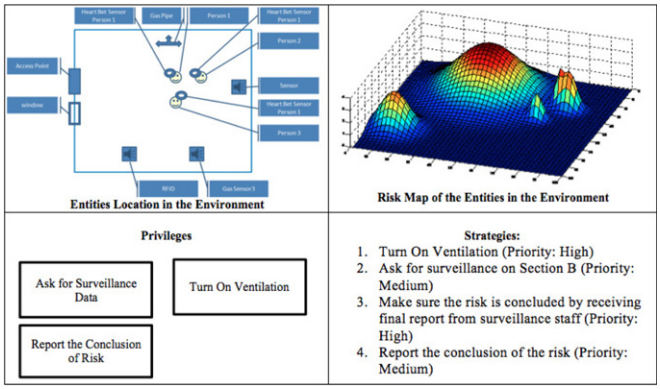

Fig. 5. An example of the *RMS* dashboard viewed by the risk manager.

Standard specifications for XACML allow the implementation of some common features of the *ACS*, and its policy model supports risk-aware features.

To avoid conflicts between rules, we adopt XACML standard combining rules. As the *ACD* is designed statically, it is possible to use these combining rules to manage conflicts at different levels. In XACML there are four main standard combining rules [22] that can be defined to construct our *ACD* and at the level of policy combining, used to avoid conflicts when *ACD* (organized in a PolicySet in XACML) apply to a certain risk(s) situation. The policy combining rules are: *Permit-Overrides*, *Deny-Overrides*, *First-Applicable*, *Only-One-Applicable* as described by Rissanen [22].

The *RMS* is implemented as a web service-based prototype that provides a dashboard in which different subjects in the risk management team can view the available data items according to their privileges that they are given by the *ACS*. They are also informed about the strategies suggested by the *RMS* as an ordered list of actions they should do to treat the risk, while the priority of each strategy is also indicated to show the level of importance of each one.

Fig. 5 shows an example of the interface of the *RMS* dashboard shown to the *Risk Manager* Subject, where he can view the risk map, and the preventive strategies with their priorities as shown on the right side of the figure. Also, using the interface, the *Risk Manager* will be able to view the monitored data about the environment and subjects and objects while being able to execute the available actions for risk treatment as his/her privileges as depicted on the left-hand side of the figure.

4.1. An example

In this section, we introduce an example showing the risk-adaptive access control model. We differentiate between global

Table 3
Examples of subjects' attributes.

General attributes			Security attributes		Geo A.
ID	Name	OrgRole	Roles	Active	Loc
RM1	John Doe	Risk Mgmt.	Risk manager Employee	No Yes	SecA
ST1	Ed Black	Surveillance	Surveillance staff Employee Security staff	Yes Yes No	SecB

Table 4
Examples of risks recognized in the environment.

Risk					
ID	Type	Level	Location	Probability	Source
Risk1	Health	Medium	SecB	High	Shov1
Risk2	Fire	Low	SecB	Low	GasP1

risks and risk for an individual. For treatment of individual risks, fine-grained calculations is done by the *RMS* to evaluate the risk-related attributes for potentially affected individuals. When dealing with a global risk that may affect a considerable number of people, these fine-grained calculations will affect the performance. Therefore, for the global risks, the *RMS* only evaluates risk-related attributes of each individual when specifically requested by the risk management team.

We consider a work environment as described in Table 1 including the different sections of the environment recognized by their *ID*; the topology (*Tp*) of the sections is described in a file, *Loc* indicates the top-right two-dimensional position of the section wrt the top-right position of the whole environment as a reference point. IncludedAreas (*IA*) represents the number of sections inside the section, *RSec* lists the reachable sections from the considered section through *Conn*, which shows the connections with other sections. The existing monitoring devices (*MD*), risk sources (*RiskS*), informative devices (*InfD*), environment protection elements (*PE*) are listed for each section. Each section has an associated sensitivity level (*SL*) and a minimum sensitivity level (*MSL*).

Now, we give examples of subjects to be protected from risk. Although these subjects can also be regarded as active Subjects that need permissions to access Objects, we consider them as passive Subjects for simplicity and only include the relevant attributes for the purpose of this example. Table 2 shows a simplified subset of attributes considered for the In-domain Subjects that are inside the environment and need to be protected. We consider also sample active subjects that need to access different physical resources and data. In our scenario, they are the Subjects in charge of risk treatment and surveillance, and of security. Table 3 shows some simplified examples of these Subjects' with a subset of relevant attributes. The roles of the subjects can either be active, meaning that they are applied in the authorization decision or not active, meaning that they are not considered at the time being.

The *RMS* monitors the entities (periodically at a given pace) and updates the attributes of the detected risks. Sample detected risks are shown in Table 4, where the *RMS* detects a Medium intensity level risk of injury with a high probability due to the presence of a risk source that is a shovel (*Shov1*) in section B. And it recognizes a low probability for the risk of fire with low intensity in section B due to the gas pipe with ID *GasP1*.

Some examples of connections between different sections are shown in Table 5 with a subset of their attributes. *AccessType [In, Out]* shows the access type for going in or out of a section from a certain connection. When no access type is set, this means that the connection is not protected and can be accessed by all subjects.

Table 5
Examples of connections of different environment sections.

ID	CType	Loc	Emg	AccessType (In, Out)
C0	In/Out	(150, 0)	No	[Badge, None]
C1	Out	(130, 120)	Yes	[None, Alarm]
C2	In/Out	(400, 450)	Yes	[Code, None]

Otherwise, there is a restriction in entering or exiting a section (e.g., *Badge* shows that a swipe card should be passed, and *Code* shows that a code should be entered to allow only certain people in or out of a section). Moreover, the *SL* and *MSL* of an In/Out connection is equal to the same attributes that are set for the sections they are entering or exiting.

Table 6 depicts examples of informative and monitoring devices such as alarms, gas detector sensors, PDA and surveillance camera, where attributes such as transmission medium and In/Out data flows are considered to characterize them.

Furthermore, we consider some Protection Elements for the Environment, Subject and Objects, such as fire protection kits, fire proof jacket, hard hat, and electricity protection gloves. Table 7 shows examples of Protection Elements and a subset of their attributes.

In Table 8 examples of tools, machineries and other risk sources are shown that can be used by the workers (Subjects). In this Table the required skills and experience needed to work safely with the tools and machineries are indicated, together with their usage instructions and the risk associated to them.

Now we consider an example:

Example 1. Consider Mary as a *Architect* who enters *section B* to check the work on the construction site. According to safety rules, since she is a medium experienced *Architect*, she can use the *Shovel (Shov1)* that has a *low* risk level associated with it. While working, the sensors in section B detect concentration of polluting substances in the air higher than 800 ppm which is out of the safe range. This is detected by the *RMS* as a risk for her health and that of other subjects in *Section B*. The *RMS* sets the values related to risk with id *Risk1* shown in Table 4. And then decides about preventive strategies as depicted in Table 9. Strategy *ST2*, which is automatic, is done by the *RMS* itself. The strategies *ST1* and *ST3* should be handled by actors. While, all the accesses to objects should be handled through the *ACS*.

In the *ACS*, access rules (rules in XACML) are defined with respect to different contexts that are defined as the targets of the *ACD* (policies in XACML). We consider samples of rules and *ACD* in what follows.

We consider $acd_1 \in ACD$ as an XACML policy having the following targets: $en.Risk.Type == \text{“}Health\text{”} \wedge en.Risk.Level \geq Medium \wedge en.Risk.Probability \geq Medium$, with the combining rule considered as *First-Applicable*. Meaning that when there is a risk in the environment *en* of type "*Health*" with higher than a *Medium* intensity level and a *Medium* level probability, then the rules in acd_1 apply in the *First-Applicable* fashion, namely, the first rule that is applicable will be considered. In the following the acd_1 together with three rules ru_1, ru_2 and ru_3:

$acd_1.ru_1$: **IF** $\{req.s.ActiveRole == \text{“}RiskManager\text{”} \wedge req.o.Type == \text{“}Ventilation\text{”} \wedge req.p == \text{“}TurnOn\text{”}\}$ **THEN** $\{effect == Permit\}$

$acd_1.ru_2$: **IF** $\{req.o.Type == \text{“}Connection\text{”} \wedge \exists\, \text{“}Evacuation\text{”} \in req.en.Risk.Strategy \wedge req.p == \text{“}GoOut\text{”} \wedge req.o.SL.Out \leq SL_4\}$ **THEN** $\{effect == Permit\}$

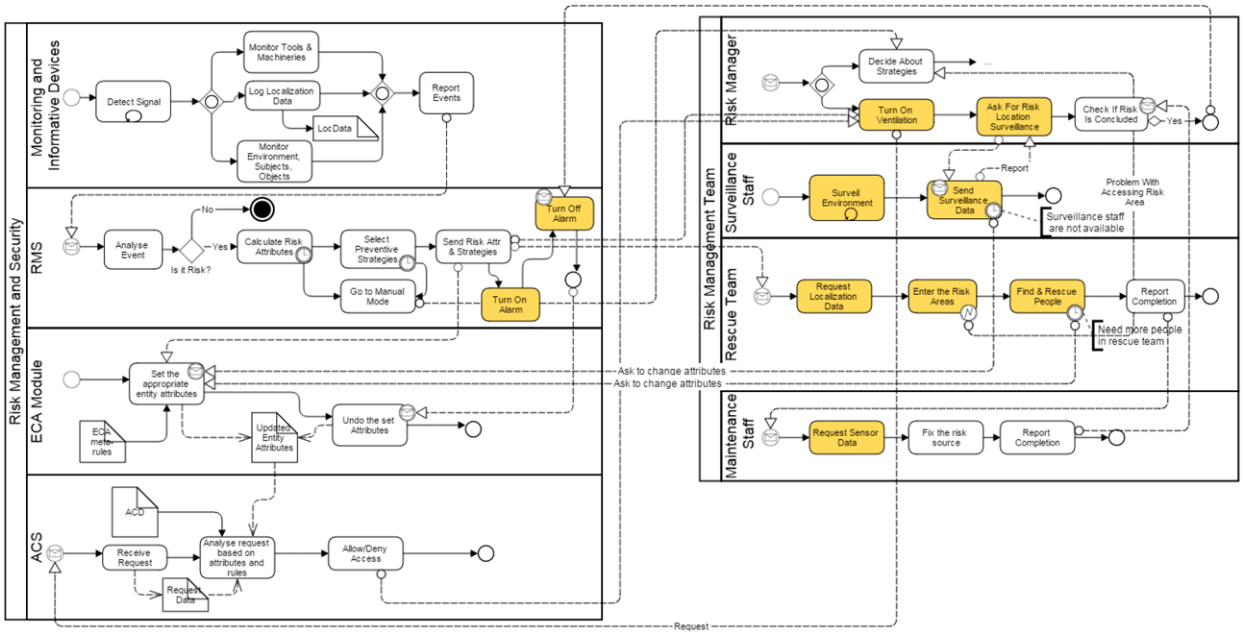

Fig. 6. Cooperation between different actors and the security and risk management systems.

Table 6
Examples of informative and monitoring devices.

ID	Device name	Safe range				
		Transmission medium	In data flow		Out data flow	
			Channel	Data type	Channel	Data type
Alarm1	Alarm	Wire	None	None	Sound	Buzzer
GasS1	Gas detector sensor	Wire	Physical	Text	None	None
PDA1	PDA	Wi-Fi	Physical	Text	Visual	Text and images
Surveil1	Surveillance camera	Wire	None	None	Visual	Text and images

Table 7
Examples of protection elements.

ID	Type	Description	Safe range		Loc
			Param.	Value	
PE1	EPE	Fire protection kit	NA	NA	SecA
PE2	IPE	Fire protection Wrap-up sheet	Temp. Impact	[0 . . . 90] °C [0. . . 1000] N	NA NA
PE3	PPE	Hard Hat	NA	NA	NA

$acd_1.ru_3$: **IF** $\{req.o.Type == Alarm \wedge$
$\exists$ "$AlarmOn$" $\in req.en.Risk.Strategy \wedge$
$req.p == $ "$AlarmOn$" $\wedge\ req.o.SL \leq SL_2\}$
THEN $\{effect == Permit\}$

Considering acd_1 and its rules, and the attributes defined for entities in the tables, the three strategies would not be permitted by the *ACS*. The *ECA* module dynamically changes the attributes of the entities based on the defined meta-rules as follows:

$meta\text{-}rule_1$: **WHEN** $ExistsNewRiskIn(R)$
IF $NotInEnvironmentRisks(r : R)$
THEN $Add(en.Risk.ID, r.ID) \wedge$
$Add(en.Risk.Type, r.Type) \wedge Add(en.Risk.Level,$
$r.level) \wedge Add(en.risk.Location, r.Location) \wedge$
$Add(en.Risk.Probability, r.Probability) \wedge$
$Add(en.Risk.Source, r.Source) \wedge$
$Add(en.Risk.Strategy, r.Strategy.Action)$

$meta\text{-}rule_1$ adds the newly notified risk attributes and the strategies to the Environment risks set. By doing so, the appropriate attributes will be set, and the rules of acd_1 will apply.

$meta\text{-}rule_2$: **WHEN** $NotEmpty(R)$
IF $Role(s) == RiskManager \wedge$
$NotExist(s.ActiveRole,$ "$RiskManager$"$)$
THEN $Set(s.ActiveRole,$ "$RiskManager$"$)$.

According to $meta\text{-}rule_2$, when any risk is notified by the *RMS*, the role of *RiskManger* for *John Doe* gets activated since his *RiskManager*role is not active as shown in Table 3. And therefore, after this change *John Doe* will be appointed as the *Risk Manager* and will have the corresponding privileges considered for this role according to the *RU*. Therefore, he will be permitted to turn on the ventilation in B according to $acd_1.ru_1$.

$meta\text{-}rule_3$: **WHEN** $ExistsNewRiskIn(R)$
IF $Equal(r.Strategy.Action,$ "$Evacuation$"$)$
THEN $Set(r.Strategy.o.SL, r.Strategy.o.MSL) \wedge$
$Set(r.Strategy.o.Conn.AccessType[In, Out], [N, N])$.

In $meta\text{-}rule_3$, it is defined that when there is a new risk notified by the *RMS* and *Evacuation* exists in the risk strategies, and there are connections in the locations affected by risk, then the sensitivity level of the connections (more precisely, the *SL* of sections that are connected) are reduced to their minimum sensitivity level and the access types are removed from the connections to let the people evacuate the affected area. This meta-rule allows the $acd.ru_2$ to evaluate to "permit".

Table 8
Examples of tools, machineries and other risk sources.

ID	Name	Skill	Exper.	Usage instruction		Relevance	ObjR	SL	LocR
				Indicator	Quantity				
Shov1	Shovel	Architect	Low	Concentration of polluting substances in the air	[0…800] ppm	High	Low	SL_1	None
PowP1	Power plug	Electrician	High	Using electric protection gloves	None	High	High	SL_3	SecC
GasP1	Gas pipes	Maintenance	Med.	Temperature near the gas pipe	Max 38 °C	High	High	SL_3	SecD

Table 9
Examples of strategies.

ID	Action	Object	Subject	Message	Priority
ST1	Evacuation	SecB	Persons in SecB	Leave the room quickly	1
ST2	Alarm on	Alarm 2	RMS	Turn on alarm 2	2
ST3	Turn on ventilation	Vent1	Risk manager	Turn on ventilation	3

meta-rule$_4$: **WHEN** *ExistsNewRiskIn*(R)
IF *Equal*(r.*Strategy*.*Action*, "*AlarmOn*") $\wedge$
GEQ (r.*Strategy*.o.SL, SL_2)
THEN *Set* (r.*Strategy*.o.SL, r.*Strategy*.o.*MSL*).

meta-rule$_4$ indicates that if a new risk action is "AlarmOn" and the Object sensitivity level is greater than SL_2 then the sensitivity is set to the minimum level allowed for that Object. This meta-rule will cause the *acd-ru*$_3$ to evaluate to "Permit" by the *ACS*.

To show the cooperation between actors in this scenario, we consider the BPM depicted in Fig. 6. On the left it is possible to view the processes of different modules including the monitoring of environment, *RMS*, *ECA* module and the *ACS*, that were explained in details in previous sections. Process activities that need authorization are shown in highlighted background, meaning that they should send a request for accessing the required resources to the *ACS* which in turn permits or denies access requests.

On the right part of the figure, the cooperative actors in the risk management team are shown, namely, the risk manager, surveillance staff, rescue team, and the maintenance staff. These Subjects cooperate with each other through the *RMS* to treat the risks. Therefore, the *RMS* recommends some strategies via the dashboard, which are visible to assigned Subjects. Then, each Subject starts conducting the instructed strategies while they collaborate with each other through messages (e.g., the risk manager can ask the surveillance staff to send him the surveillance data, which needs the authorization of the *ACS*, or the rescue teams to report the completion of their work, so the maintenance staff can start fixing the risk source).

In case some staff members are not available, or need more people to help for completion of their tasks, a request can be sent to the *ECA* module, that will change the entity attributes according the predefined meta-rules, to adapt the *ACS* to allow access to the required resources to the people who in normal situations would not have such permissions. As an example, if the rescue team are short in staff and need more persons to "find and save people in risk", they can send a request to the *ECA* module to add some of the trained staff to the rescue team.

As shown in *meta-rule*$_5$, when there exists a risk in R and a request is received by the *ECA* module stating that the rescue team is short in staff, if there are trained staff people to join the rescue team, the *ECA* module will add the "RescueTeam" role to their roles and activate it. In this way the trained staff can join the rescue team.

When the maintenance staff completes fixing the risk source, it notifies to the risk manager who will check the successful conclusion. In the positive case, the *RMS* will turn off the alarm, and will notify the *ECA* module to undo the changes in entity attributes that were made during the risk.

meta-rule$_5$: **WHEN** *ExistsRiskIn*(R) $\wedge$
ReceivedRequest (*param*)
IF *Equal*(*param*, "*ShortinRescueStaff*") $\wedge$
NotEmpty(*TrainedRescueStaff*)
THEN *Add*(*TrainedRescueStaff* .*Role*, "*RescueTeam*") $\wedge$
Set (*TrainedRescueStaff* .*ActiveRole*, "*RiskManager*").

5. Related work

In risk management, the issues of risk treatment and of providing help to people affected by the risk, once it is recognized through surveillance devices, is an open issue [23]. Recent advances in risk and disaster detection and management technologies and ICT support infrastructures have enabled the generation and reliable deliveries of machine-readable early disaster warnings over all communication pathways. Some approaches have been developed to preserve security and privacy during risk treatment procedures [24]. Liu et al. [25] advocates the development and pervasive deployment of intelligent and secure guards against disasters. Considering smart devices, applications and services are capable of authenticating standard risk warning and response messages from authorized senders and of appropriate actions to help people stay safe. In such streamline, based on our previous research on risk management in work areas (see [7]), our work addresses a system for managing and treating risks and authorizing cooperation between different actors to conduct strategies for risk prevention while respecting the security rules. With respect to other works, our research focuses on proposing a risk-aware access control model facilitating the cooperation for risk. With the emergence of the concept of smart environments, security needs be properly addressed. While a great amount of data is sensed from an urban environment for risk management, employing a risk-adaptive security model that allows collaboration in risk-management is critical [26]. Therefore, proper access control mechanisms must be in place for maintaining privacy and security to avoid unauthorized access to data, and to prevent users, processes or applications to misuse data in smart spaces [27].

For security models, there has been considerable interest in Attribute Based Access Control (ABAC) [8] due to the limitations of models such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role-Based Access Control (RBAC) [28]. ABAC can successfully encompass MAC, DAC, and RBAC limitations while attributes of subjects, objects, and the environment allow fine-grained and flexible access control and facilitate the dynamic adaptation of the security rules that is necessary in a risk prone environment. We propose using ABAC as the basis of our risk-aware access control model, while extending it to dynamically

adapt to the risks that are recognized in the environment to allow the efficient risk treatment.

The works in risk-aware access control models focus on the concept of risk-awareness in the view of managing the trade-off between the risk of allowing an unauthorized access, and the cost of not permitting it [29]. While we draw the focus on adapting the model to have a fine-grained access control considering several risk attributes calculated by the risk management system.

Many researches tackle the problem of risk awareness based on RBAC [29–32]. Chen and Crampton [29] proposes a risk-aware access control model based on RBAC where the debate is about the risk of allowing access, based on factors such as: the user trustworthiness, the degree of competence wrt the assigned roles, and the degree of appropriateness of a permission-role assignment for a given role. However, they do not consider the role of risk mitigation strategies in assigning the required permissions needed to successfully conclude the risk treatment. Venkatasubramanian et al. [32] introduces Critically Aware Access Control (CAAC) as an adaptive access control mechanism for emergency management in smart environments based on RBAC. Considering the health problems of a person (like falling), the access control system dynamically makes changes in the privileges to let him/her receive the necessary care by people who can access his/her sensitive data. Yet, the paper does not consider adaptations of access control rules to manage a more general case in which an environmental risk (like fire) occurs and complex strategies need to take place through the presence of cooperating actors.

The issues of these approaches, appearing for instance in [29–32] to be used for environmental risk and emergency management, is twofold. First, in an emergency situation, quick response of the access control system is essential, and computing the trust, user competence and similar functions for each access and for each user can be a considerable overhead. Furthermore, RBAC suffers many limitations that make it insufficient for many applications [33].

Other approaches such as Petritsch [34] make use of break-glass policies for exception-handling mechanisms and to allow for controlled overwriting of access control rules. However, using break-glass policies, fine-grained definition of the access control rules for different situations (contexts) is not possible since contexts are considered to be only emergency and non-emergency.

A similar approach [35] tries to overcome some limitations of RBAC by introducing the concept of attributes on RBAC for risk-based adaptive access control (RAdAC). This provides a dynamic balance between the need to access information in view of mission priorities, risk and cost of information compromise, and overall operational and threat status of the system. In this work, the authors consider local and global situational factors regarding what happens in the environment as an attribute; however, they do not introduce any methods to evaluate and handle the risk nor the tool architecture and implementation issues. More recently, Smari et al. [9] extends the ABAC model to incorporate trust and privacy issues in cross-organizational collaborative crisis management. Many issues are treated, such as network security for communications during the crisis, and trust in collaborations. Our work is similar in that we also consider the situational factors for adaptiveness of the access control model, such as events in the environment connected to time and location of the access. We choose the ABAC model to represent the dynamicity. However, we focus on keeping the security changes compliant with overall organizational policies using a confinement property through Access Control Domains. Moreover, in a risk situation, adaptivity should be highly automatic, even without human confirmation, possibly requiring only an immediate automatic decision or a suggested strategy of intervention to be communicated to human responsible and teams. Therefore, we focus on capturing events and identifying risks so as to rule out the access controls according to the risk to prevent its occurrence.

Feng and Zheng [36] treats cooperation between organizations and the flexible exchange of security information across organizations. The focus is on effectively managing information systems security in a distributed environment. A cooperative model for security risk management in a distributed environment is proposed supported by Bayesian networks. Our cooperative approach works in a similar way and inherits many of these concepts. It elaborates on people and tools cooperation. Critical infrastructure systems considered as complex elements forming networked systems of systems are discussed in [37]. Risk is considered the product of three complex and interrelated elements: threats, vulnerabilities, and consequences. An approach similar to our ECA approach is undertaken. However, the scope is different in that it addresses disasters and disruptions, such as global climate changes. The focus is on decision support tools for analyzing and actively managing risk. Tarrant et al. [38] consider the Web as an increasingly relevant platform for linked data, including risk-related data such as those describing a crisis. An example is given for file format registries in the evaluation of risks. Here the requirement for information shared among various institutions in gathering and collating information is the focus. The interesting part is how the Web can promote recognizing different formats of data from multiple sources and how a registry and its services can be constructed as a reference platform to allow and encourage publication of preservation data. These are aspects that we aim to address in our near-future research.

6. Concluding remarks

This paper has presented risk and dynamic access control in risk-prone environments, taking into account cooperation and issues of data representation and sharing during risk management. Risks are approached from a preventive perspective. They are recognized by the *Risk Management System-RMS* based on the monitoring data acquired from the environment. The *RMS* produces, as an output, a detailed description of risk and of the strategies suitable to prevent it. The phases have been illustrated through which the *RMS* processes events, risks and strategies, and outputs security rules modifications passed on to the *ACS* portion of the system architecture for risk management. Based on the ABAC paradigm and using the XACML policy language, we have proposed a dynamic access control model which is adaptive to risks in that access control rules are modified dynamically for subjects and objects under temporary grant/revoke operations. To foster adaptivity, we introduced the notion of *Access Control Domain* delimiting the scope of dynamic authorizations of subjects to access objects in a cooperative way on the basis of predefined access control policies that regulate the modification of security rules. We have presented a sample scenario and the processing performed by the *RMS* to handle risks and to output the adaptive security needs. Future work includes extending these ideas through research on data representation and sharing, on efficiently viewing the data items that are needed for effective risk treatment, and on facilitating the peer-to-peer collaboration between risk team members. We will focus on a refinement of the model to handle subject groups, which can help to cluster subjects with homogeneous privileges and needing to cooperate, while considering the disjunction among groups so that the need-to-know principles would not get violated when belonging to different groups simultaneously. We will also consider uncertainty in detection of risks and how to handle or decrease false alarms and the cost of having such situations.

Acknowledgments

This work has been performed in the frame of the Italian Projects *Sensori* and *Attiv@bili*. The work is partially funded by the EU Project *ECO2Clouds* (grant 318048). We thank Prof. Emil Lupu of Imperial College for valuable suggestions about security policies management.

References

[1] C. Dobre, F. Xhafa, Intelligent services for big data science, Future Gener. Comput. Syst. 37 (2014) 267–281.
[2] M. Rönkkö, J. Heikkinen, V. Kotovirta, V. Chandrasekar, Automated preprocessing of environmental data, Future Gener. Comput. Syst. 45 (2014) 13–24.
[3] D.J. Weitzner, J. Hendler, T. Berners-Lee, D. Connolly, Creating a policy-aware web: Discretionary, rule-based access, in: Web and Information Security Vol. 1, 2006.
[4] R.S.H. Piggin, Governance, risk and compliance: impediments and opportunities for managing operational technology risk in industrial cyber security and safety, In: System Safety and Cyber Security (2014), The 9th International Conference on System Safety and Cyber Security Conference, Session 4b: Cyber Physical Systems, 15 and 16 October 2014, Manchester, pp. 1–8.
[5] E. Borodzicz, Risk, Crisis and Security Management, Wiley, 2005.
[6] H. Abbas, C. Magnusson, L. Yngstrom, A. Hemani, Addressing dynamic issues in information security management, Inf. Manag. Comput. Secur. 19 (1) (2011) 5–24.
[7] M. Fugini, C. Raibulet, L. Ubezio, Risk assessment in work environments: modeling and simulation, Concurr. Comput.: Pract. Exper. 24 (18) (2012) 2381–2403.
[8] V.C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, Guide to attribute based access control (ABAC) definition and considerations, NIST Spec. Publ. 800 (2014) 162.
[9] W.W. Smari, P. Clemente, J.-F. Lalande, An extended attribute based access control model with trust and privacy: Application to a collaborative crisis management system, Future Gener. Comput. Syst. 31 (2014) 147–168.
[10] G. Danezis, Trust as a methodological tool in security engineering, in: Trust, Computing, and Society, Vol. 68, 2014.
[11] M. Hanspach, J. Keller, In guards we trust: Security and privacy in operating systems revisited, in: Social Computing (SocialCom), 2013 International Conference on, IEEE, 2013, pp. 578–585.
[12] M. Fugini, G. Hadjichristofi, M. Teimourikia, Adaptive security for risk management using spatial data, in: Database and Expert Systems Applications, Springer, 2014, pp. 343–351.
[13] E. Paja, P. Giorgini, S. Paul, P.H. Meland, Security requirements engineering for secure business processes, in: Workshops on Business Informatics Research, Springer, 2012, pp. 77–89.
[14] G. Purdy, Iso 31000: 2009—setting a new standard for risk management, Risk Anal. 30 (6) (2010) 881–886.
[15] P. Bonato, Wearable sensors and systems, IEEE Eng. Med. Biol. Mag. 29 (3) (2010) 25–36.
[16] D. Garlan, B. Schmerl, S.-W. Cheng, Software architecture-based self-adaptation, in: Autonomic Computing and Networking, Springer, 2009, pp. 31–55.
[17] M. Fugini, C. Raibulet, L. Ubezio, Risk characterization and prototyping, in: New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference on, IEEE, 2010, pp. 57–64.
[18] H.F. Tipton, M. Krause, Information Security Management Handbook, CRC Press, 2012.
[19] F. Casati, S. Castano, M. Fugini, Dynamic task assignment in workflows. US Patent 7,155,720, December 26, 2006.
[20] N. Dessì, M. Fugini, G. Garau, B. Pes, Architectural and security aspects in innovative decisional supports, in: Proc. ITAIS'13 Conf, 2013.
[21] W. Balana, Balana: Open source xacml implemantation, 2014. https://github.com/wso2/balana.
[22] E. Rissanen, eXtensible access control markup language (XACML) version 3.0 OASIS standard, 2012.
[23] K. Smith, Environmental hazards: assessing risk and reducing disaster, Routledge, 2013.
[24] N. Poolsappasit, R. Dewri, I. Ray, Dynamic security risk management using Bayesian attack graphs, Dependable Secur. Comput. 9 (1) (2012) 61–74.
[25] J.W.-S. Liu, C.-S. Shih, E.T.-H. Chu, Cyberphysical elements of disaster-prepared smart environments, IEEE Comput. 46 (2) (2013) 69–75.
[26] L.M. Camarinha-Matos, H. Afsarmanesh, Collaborative systems for smart environments: Trends and challenges, in: Collaborative Systems for Smart Networked Environments, Springer, 2014, pp. 3–15.
[27] A. Evesti, J. Suomalainen, E. Ovaska, Architecture and knowledge-driven self-adaptive security in smart space, Computers 2 (1) (2013) 34–66.
[28] X. Jin, R. Krishnan, R. Sandhu, A unified attribute-based access control model covering DAC, MAC and RBAC, in: Data and Applications Security and Privacy XXVI, Springer, 2012, pp. 41–55.
[29] L. Chen, J. Crampton, Risk-aware role-based access control, in: Security and Trust Management, Springer, 2012, pp. 140–156.
[30] D.H. Huang, Y.Q. Yang, Role-based risk adaptive access control model, Appl. Mech. Mater. 416 (2013) 1516–1521.
[31] K.Z. Bijon, R. Krishnan, R. Sandhu, A framework for risk-aware role based access control, in: Communications and Network Security (CNS), 2013 IEEE Conference on, IEEE, 2013, pp. 462–469.
[32] K.K. Venkatasubramanian, T. Mukherjee, S.K. Gupta, CAAC—an adaptive and proactive access control approach for emergencies in smart infrastructures, ACM Trans. Auton. Adapt. Syst. (TAAS) 8 (4) (2014) 20.
[33] R. Sandhu, The authorization leap from rights to attributes: maturation or chaos? in: Proceedings of the 17th ACM symposium on Access Control Models and Technologies, ACM, 2012, pp. 69–70.
[34] H. Petritsch, A generic break-glass model, in: Break-Glass, Springer, 2014, pp. 37–50.
[35] S. Kandala, R. Sandhu, V. Bhamidipati, An attribute based framework for risk-adaptive access control models, in: Availability, Reliability and Security (ARES), 2011 Sixth International Conference on, IEEE, 2011, pp. 236–241.
[36] N. Feng, C. Zheng, A cooperative model for is security risk management in distributed environment, Sci. World J. 2014 (2014).
[37] J.C. Cummings, B. Holtz, M. Riddle, D. Ullman, Modeling and simulation to support risk management in complex environments, 2014, available at http://c4uc.org/Portals/2/Papers/TS3A_Mod-Sim-Risk-Mgmt_Paper_JCummings.pdf.
[38] D. Tarrant, S. Hitchcock, L. Carr, Where the semantic web and web 2.0 meet format risk management: P2 registry, Int. J. Digit. Curation 6 (1) (2011) 165–182.