Leveraging Smart Environments for Runtime Resources Management

Paolo Barsocchi, Antonello Calabró(✉), Francesca Lonetti, Eda Marchetti, and Filippo Palumbo

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche (CNR), via G. Moruzzi 1, 56124 Pisa, Italy
{paolo.barsocchi,antonello.calabro,francesca.lonetti, eda.marchetti,filippo.palumbo}@isti.cnr.it

Abstract. Smart environments (SE) have gained widespread attention due to their flexible integration into everyday life. Applications leveraging the smart environments rely on regular exchange of critical information and need accurate models for monitoring and controlling the SE behavior. Different rules are usually specified and centralized for correlating sensor data, as well as managing the resources and regulating the access to them, thus avoiding security flaws. In this paper, we propose a dynamic and flexible infrastructure able to perform runtime resources' management by decoupling the different levels of SE control rules. This allows to simplify their continuous updating and improvement, thus reducing the maintenance effort. The proposed solution integrates low cost wireless technologies and can be easily extended to include other possible existing equipments. A first validation of the proposed infrastructure on a case study is also presented.

Keywords: Smart environment · Monitoring · Sensors
Access control policy

1 Introduction

The interaction between people and ubiquitous computing [1], nowadays identified as a Smart Environment (SE), is an important area of interest. The SE paradigm depends on communication and cooperation among numerous devices, sensor networks embedded in the environment itself, servers in a fixed infrastructure and the increasing number of mobile devices carried by people. Thousands of smart devices are now operating in cities, gathering information and providing smart applications for e.g. environmental monitoring, energy management, traffic management, smart buildings and smart parking [2]. These devices integrate computation, networking, and physical processes and are able to monitor and control physical objects providing an extremely efficient and economic mean for improving the quality of life of citizens.

D. Winkler et al. (Eds.): SWQD 2018, LNBIP 302, pp. 171–190, 2018.
https://doi.org/10.1007/978-3-319-71440-0_10

From the networking point of view, one of the solutions adopted by the SE is constituted by the Wireless Sensor Networks (WSNs) [3], which are autonomous devices managing sensing, computing and wireless communication capabilities. The integration of computation, networking, and physical processes require regular exchange of critical information in timely and reliable fashion and a specific modeling and control of the SE behavior. Considering, in particular, the control point of view, a SE usually involves three levels of rules: (i) the rules for managing and correlating sensors data and technologies; (ii) the rules able to guide the SE process, to define the users and the systems behavior, and to protect against possible problems and inconveniences faults; (iii) the access control rules that manage the resources (sensors, technologies, data, and so on) and protect against possible malicious use or security flaws.

However, independently of the formalism adopted, writing such kind of rules is a hard, verbose and error-prone activity. Considering in particular behavioral and access control ones, their modifications and updating may cause inconsistencies and/or security flaws; therefore, an accurate, time and effort consuming validation and verification should be implemented [4, 5].

Especially in large scale organizations, in order to partially solve this problem, the common practice is to exploit the internal regulations and network specification requirements so to define just the basic control rules that may remain unchanged for a considerable amount of time. Thus existing solutions, generally try to adopt a static specification of the different rules and to centralize their control inside the architecture. As side effect the management of SE behavior could become quickly outdated over time, leading either inconsistencies with the evolved behavioral and technological organization environment, or security vulnerabilities.

From the previous considerations, the solution proposed in this paper relies on the decoupling the different levels of control rules, so to maximize their effectiveness and reduce as much as possible their maintenance and updating effort. The control levels considered are: the *Sensors Rules*, which define the sensors behavior and activities; the *Usage Control Rules*, which define the users and sensors interactions; and the *Access Control Rules*, which manage the accesses to the different resources expressed through a specific control policy formalism. As better detailed in the remaining of this paper, from a practical point of view, each of the control rule levels is in charge of a different independent reasoner, in order to completely separate the continuous behavioral and accesses control from physical network control.

The paper is organized as in the following. Section 2 highlights more the motivations of the proposal and the research questions considered. Section 3 presents some basic concepts about usage and access control systems. Section 4 presents the proposed infrastructure for runtime management and control of smart environments. Section 5 provides a first validation of the proposed approach on a case study. Related work are presented in Sect. 6 whereas Sect. 7 concludes the paper also hinting at future work.

2 Motivations and Research Questions

Much of the research work has been done in the field of smart buildings, smart home, smart city, however there is still the necessity of improving their quality and performance in the management of control rules (Sensors Rules, Usage Control Rules and Access Control Rules) so to make them smarter in taking intelligent and prompt decisions [6]. In the field of rule based systems, some smart proposals have been developed, especially in processing schemes [7], which mainly consist of algorithms and the complex event processing mechanism. However for reducing the overall delivery, operation and maintenance effort of the control rules typically a static approach is adopted: i.e. fixing (or rarely varying) the control rules and directly embedding them into the processing engine controlling the environment so to minimize their verification and assessment.

The target of this paper is enhance the flexibility and adaptability of the current state of the practice, by proposing a solution in which the different control rules or (subsets of them) can be activated/inhibited or on-the-fly defined and modified according to runtime organization exigencies. The idea is to leave the freedom to the organization manager to select or define the sets of control rules more suitable for his/her purpose at any specific time. For this the processing engine controlling the smart environment is enhanced with features for either directly specifying the proper set of control rules or to select the most suitable ones from a pre-defined collection of frequently adopted rules. A dedicated engine will manage the frequent updates/modifications of the set of rules, checking their correctness and compliance, and overriding when necessary the those anymore valid, without an impact on the overall management of organization.

The adoption of automated infrastructure for the definition and assessments of control rules is a valid help in their specification and implementation. Moreover it represents an important improvement for quality and performance of the smart environment, because it enhances the control of the possible violations or security problems. Summarizing the proposal of this paper will improve the current state of the practice by:

- providing the possibility to modify and implement in extremely short time, specific and temporary rules. For instance manager will have the possibility to automatically change the access control rules of a specific environment many times during a day (or particular period), in relation to either the role of the subject requiring the access, or the access time or the sensor values available for the considered environment.
- Forcing the persons in charge of control rules to formally define them for any critical situations. This avoids the specification of generic control rules which could be inefficient or even useless in case of problematic conditions. This will have several positive impacts: (i) focus on control rules on specific problems, so minimize misunderstanding or weaknesses in their implementation and management; (ii) provide more suitable solutions due to the possibility to early define or adapt the corrective actions to be performed in case of problems, security or safety flaws; (iii) better document the organization control

behavior in case of specific (critical) situations. Indeed many times specific control rules are just best practices inside an organization and not formally defined. Exploiting automatic facility for the definition and collection of control rules provides a useful knowledge database that can be used in case of similar situations or for training new personnel.
– The separation of the different control rules into three levels (Sensors Rules, Usage Control Rules and Access Control Rules) so to better map the distribution of knowledge between organization stakeholders. The use of a unique infrastructure for the definition, management and implementation of the three different levels of control rules lets the personnel with different, and many time separated, background knowledge to easily interact and work together in a unique environment. This from one side will keep the separation of roles and knowledge and from the other will improve the overall organization control correctness.

In the developing of the proposed infrastructure the following research questions have been considered:

– RQ1: Flexibility: is the proposal useful for improving the definition of control rules for different specific situations?
– RQ2: Adaptability: is the proposal useful for improving the selection of the proper control rules depending on specific environmental conditions?
– RQ3: Low cost: is the overhead introduced by the proposed infrastructure acceptable? In particular, we will assess whether it can have an impact on the overall cost of smart environment implementation, installation and integration.
– RQ4: Usability: is the proposed usable by not domain experts? In particular, we will assess whether the required technical baseline knowledge can have an impact on usability of the proposed implementation.

3 Access and Usage Control

Access control is one of the most adopted security mechanisms for the protection of resources and data against unauthorized, malicious or improper usage or modification. In the last decades, a large variety of policies and policy models have been proposed to define authorized operations on specific resources, such as (RBAC) Model [8] and (XACML) [9]. An XACML policy defines the access control requirements of a protected system. An access control request aims at accessing a protected resource in a given system whose access is regulated by a security policy. The request is evaluated on the Policy Decision Point (PDP) against the policy and the access is granted or denied.

A simplified vision of an XACML policy has a hierarchical structure: at the top level there is the policy set, which can contain in turn one (or more) policy set(s) or policy elements. A policy set (a policy) consists of a target, a set of rules and a rule combining algorithm. The target specifies the subjects, resources, actions and environments on which a policy can be applied. If a request satisfies

the target of the policy set (policy), then the set of rules of the policy set (policy) is checked, else the policy set (policy) is skipped. A rule is composed by: a target, which specifies the constraints of the request that are applicable to the rule; a condition, which is a boolean function evaluated when the request is applicable to the rule. If the condition is evaluated to true, the result of the rule evaluation is the rule effect (*Permit or Deny*), otherwise a *NotApplicable* result is given. If an error occurs during the application of a request to the policy, *Indeterminate* is returned. The rule combining algorithm specifies the approach to be adopted to compute the decision result of a policy when more than one rule may be applicable to a given request. Listing 2 provides an example of XACML policy.

Usage control model (UCON) [10] is one of the emerging and comprehensive attribute based access control models that has the ability of monitoring the continuous updates in a system addressing the limitations related to attribute mutability and continuous usage permission validation. UCON model is an extension of the traditional access control models, which besides authorizations introduces new factors in the decision process, namely obligations, conditions, and mutable attributes. Mutable attributes are paired with subjects and objects, and their values are updated as a consequence of the decision process. Hence, the attributes that have been evaluated by the security policy to initially grant an access to a resource could change their values, while the access is in progress in such a way that the access right does not hold anymore. In this case, the access should be interrupted to preserve the system security. For this reason, UCON policies specify whether or not a decision factor must be evaluated before and/or during the usage of the resource (continuous policy enforcement).

4 Infrastructure

In this Section, we provide some details about the infrastructure used to decouple Sensors, Usage Control and Access Control rules in order to reply also to research questions of Sect. 2.

As shown in Fig. 1, the Infrastructure is conceptually divided in different nodes (see Fig. 1): (i) the *Access Control Engine* is the node in charge of implementing the access control management. (ii) the *Glimpse: Monitoring Infrastructure* is the node monitoring and enforcing the Sensors and Usage rules; (iii) the *Sensors and Actuators* are physical (hardware) components of the infrastructure: (iv) the *Management Interface* is the GUI (Graphical User Interface) through which the different rules can be defined and feedbacks and log analysis can be provided.

As in Fig. 1, the Administrators are in charge of providing the definition of the three levels of rules for the overall infrastructure. This can be done by means of a *GUI* on which *Rules editor* and *Policies editor* components are running. Specifically, through *Rules Editor* the Administrators can define the Sensors and Usage rules using a specific language (further details are provided in Sect. 4.2). Additionally, by means of *Policy Editor* they can define the XACML access control policies that will rule the resources access. Finally, through the *GUI*

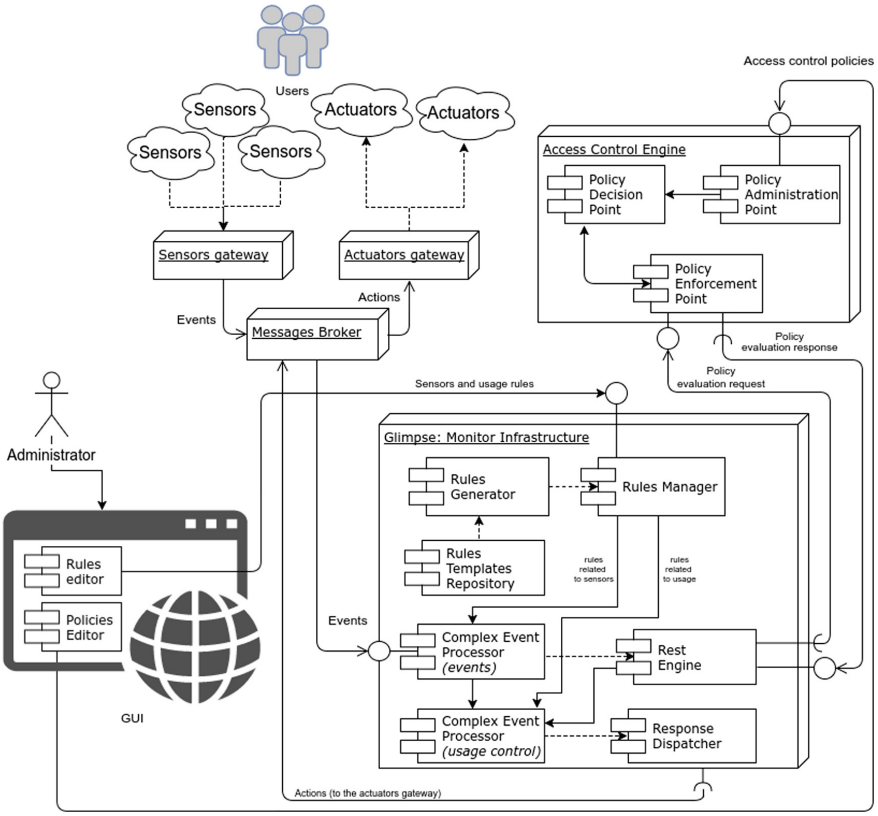

Fig. 1. Proposed architecture

the Administrators can visualize logging data, monitoring results, sensors and actuators status. In the following subsections, more details about the above mentioned nodes are provided.

4.1 Access Control Engine

This node manages the resource access by enforcing the XACML Policy defined by the Administrators. In particular, the *Access Control Engine* node contains three components (Fig. 1 top right): (i) the *Policy Enforcement Point (PEP)*, usually embedded into an application system. It receives the access request in its native format from the *Glimpse: Monitoring Infrastructure*, constructs an XACML request and sends it to the *Policy Decision Point (PDP)*; it receives the PDP responses and forwards them to the *Glimpse: Monitoring Infrastructure* through its REST (REpresentational State Transfer) Interface called *REST Engine*; (ii) the *Policy Decision Point (PDP)* evaluates the policy with respect to the request and returns the response, including the authorization decision to

the *PEP*; (iii) the *Policy Administration Point (PAP)* is the component entity in charge of managing the policies and deploying them on the PDP; it receives the XACML access control policy by the *Management Interface*.

4.2 Monitoring Components

The monitor infrastructure, integrated into the proposed infrastructure, is a flexible, adaptable and dynamic solution independent of any specific sensor or access control network notation or execution. With respect to similar components and tools currently available, the monitor infrastructure included in this proposal, has been enhanced with facilities for activation the counter measures or the recovering activities in case of violations of some performance constraints. These constraints are not mandatory specified at the system startup, but can be automatically raised from the rule engines involved or can be improved at runtime by injecting new rules on the complex event processors. The monitoring framework presented in this paper has been inspired by the monitoring architecture presented in [11,12]. The *Glimpse: Monitoring Infrastructure* node (Fig. 1) manages the complex event processing and the interactions with *Sensors*, *Actuators* and *Access Control Engine*, and includes new features devoted to the usage and access control request generation.

The main monitoring components are:

- The *Rules Manager* component is in charge of orchestrating the rules generation starting from the templates stored within the component *Rule templates Repository* through the *Rules Generator* component.
- The *Rules Generator* is the component in charge of synthesizing the rules starting from the directives received by the *Rules Manager* by means of techniques based on generative programming approaches [13,14].
- The *Rules Templates Manager* is an additional internal repository storing the meta-rules enabling the run-time adaptation by means of generative procedures.
- The *CEP - Events* CEP (Complex Event Processing) Events is a rule engine realized by means of the Drools rule language [15]. It correlates the events flowing from *Sensors* with the rules loaded by the *Rules Manager* component.
- The *CEP - Usage* is in charge of correlating complex events generated by the *CEP - Events* with the rules related to the usage of the resources, loaded by the *Rules Manager*.
- The *Rest Engine*, is the component in charge of communicating through REST [16] interfaces with the *Access Control Engine* in order to send/receive the *Access Control Engine* request/response.
- The *Response Dispatcher* through the *Message Broker (AMQ)*, realized by means of ActiveMQ [17], sends events to the actuators managed by the *Actuators gateway*.

The peculiarities of the proposed architecture is to include for the first time a chain of two CEP entities, the *CEP - Events* and the *CEP - Usage*, for decoupling

the activities concerning the management of the sensors from those more related to the administration of the resource usage and alarming situations. This makes easier the definition of new primitive events generated by (new/updated) sensors and the inferring of events in the form of composite events in a way completely independent of the access and usage control rules. Moreover, it lets a quick and high level updating of the general resource access and usage regulations and the planning of specific corrective actions in case of alarms or resource violations, leveraging from the specific sensor network on which they are implemented.

From a practical point of view, all the communications among monitoring components are performed through messages sent to the *Message Broker (AMQ)*, running on top of an Enterprise Service Bus like ServiceMix [18]. In order to improve the communication security, each component exposes a SSL certificate (self-signed certificates).

4.3 Sensors and Actuators Components

Sensors and *Actuators* (in top left side of Fig. 1) are deployed over the network and communicate with the infrastructure through the *Sensors gateway* and the *Actuators gateway*, respectively. These hardware components send messages to the *Glimpse: Monitoring Infrastructure* through a *Message Broker (AMQ)* using a predefined event type.

From a technical point of view, the sensor nodes considered in the proposed infrastructure are based on the module Core2530, produced by WaveShare Electronics[1]. The sensor nodes are connected with a ZigBee node[2], which is able to: evaluate the real power consumption of the target environment; identify all the indoor movements of users through Passive Infrared Sensors (PIR-based motion sensors); detect environmental noise; measure temperature and relative humidity. Every node of the distributed sensor network is configured as a ZigBee router to exploit multi-hop functionality and it periodically sends the measured data to the ZigBee coordinator. The ZigBee network ensures a reliable communications in indoor environments without suffering due to the multipath effect [19]. Moreover, each user is equipped with a Bluetooth Low Energy (BLE) beacon[3], which periodically sends a message useful for locating and identifying the user. Figure 2 shows the hardware used for sensing the environment. In particular, on the left there are the RadBeacon Dot[4] and the BLED112[5] used as a sender and receiver beacon, while on the right side there is a ZigBee node.

The middleware, named *Sensor Weaver*, uses ZB4O[6] to interact with sensors and actuators deployed in the ZigBee networks [20] and integrates the BLE in

[1] http://www.wvshare.com/.
[2] http://www.zigbee.org/.
[3] https://developer.mbed.org/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/.
[4] http://store.radiusnetworks.com/collections/all/products/radbeacon-dot.
[5] https://www.bluegiga.com/.
[6] http://zb4osgi.aaloa.org/.

Fig. 2. The hardware used to sense the environment

order to abstract the different kinds of technologies. The gateway node provides access to the sensors discovered through an IP network and a communication platform. The main goal of Sensor Weaver is to provide a secure communication platform for the exchange of sensing information in a distributed sensor network environment [21,22]. Moreover, Sensor Weaver also provides tools and applications that enable long-term sensor monitoring and automatically control the actuators deployed in the WSN [23,24].

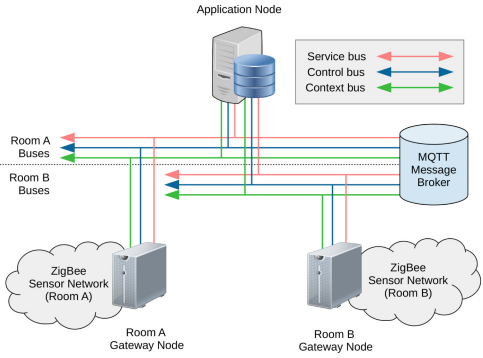

Fig. 3. The architecture of sensor weaver

In order to separate communication concerns of Sensor Weaver, we designed several communication buses. Each bus has a specific managing role (see Fig. 3): (i) a *Service Bus* for the service life-cycle events; (ii) a *Context Bus* for the sensor measurement updates; (iii) a *Control Bus* for the invocations of actuators. We implement Sensor Weaver on top of the OSGi platform and we use the MQTT messaging service [25,26].

4.4 Research Questions Analysis

As evidenced by the technical description of the infrastructure, its development has been focused on the improving as much as possible its flexibility and adaptability. In particular the availability of the *Management Interface* makes easier

the definition of different kinds of control rules, lets to target on specific situations and provides the visualization of the monitor results. Moreover the availability of editors, such as the *Rules editor* and *Policies editor*, let a more friendly and usable definition of the control rules especially for people not expert in the different specification languages. All this evidences positively reply to the RQ1 and RQ4 of Sect. 2.

Concerning instead the RQ2, adaptability is guaranteed by the separation of the infrastructure into the different nodes (see Fig. 1), each one responsible of the implementation and management of separate set of control rules. The facilities provided let to detect and to correlate events generated by different layers supporting in parallel sensors, access and usage control features. In particular the innovate adoption of a chain of two CEP entities, assures the decoupling of the activities concerning the management of the sensors from those related to the resource usage. This let also a better management of alarming conditions and maximizing the adaptability of the infrastructure to different situations and exigencies.

Finally considering the RQ3, as highlighted in Sect. 4.3, cost of the proposed infrastructure is mitigated by the choice of the technology adopted for the infrastructure implementation. Indeed among the different proposals, the trade-off solution adopted in this paper relies on ZigBee [27]. This is a recognized low cost standard-based wireless technology designed to address the unique needs of low-power WSNs, and to model the different resource capabilities. It guarantees the non-invasiveness of the installations and the possibility of integration with other possible existing equipments.

5 Infrastructure Application

In this section, we describe the usage of the proposed infrastructure for the management of a cold storage room inside the *ISTI-CNR* (Istituto Scienza e Tecnologie dell'Informazione - Consiglio Nazionale delle Ricerche of Pisa Italy)[7] research area. However due to space limitation sand for aim of simplicity, we report here just a simplified description of the management of this medical cold storage called *Laboratory Cold Room (LCR)*. It is out of the scope of this paper to go into the complex details of the sets of rules necessary for managing the LCR. Here, we voluntarily keep the scenario simplified to better explain the role and the advantages of the proposed infrastructure. The complete description of the implementation can be found in [28].

The control of Laboratory Cold Room focuses on three different main aspects: to keep the temperature required for safe and secure storage of a wide range of laboratory materials; to rule the access to the room; to activate corrective actions in case of detected violations or alarming situations. For this the room has been instrumented with different sensors and several sets of control rules have been defined. These last include: (i) Sensor Rules for managing the security boundary value of each sensor or combination of them (for instance, the tolerance

[7] http://www.isti.cnr.it.

temperature, humidity ranges, and so on); (ii) Access Control policies ruling who and when can access LCR (name or the role of people that are allowed to work in the LCR); (iii) Usage Control Rules for managing sensors failures, resource violations, and alarming situation in general (for instance, the technical personnel to be called in case of problems).

5.1 Case Study Set up

As shown in Fig. 4 the Laboratory Cold Room has been equipped with a beacon receiver for gathering data from BLE beacons and with several sensors (see Sect. 4.3 for more details), which are: Temperature and humidity; Presence (PIR-based); Energy consumption; Noise detector; RFID Bagde Reader for Room access Control. An instance of the *Monitoring Infrastructure* has been deployed on: *ubuntu@n037.smart-applications.area.pi.cnr.it*, a virtual machine running on top of ISTI-CNR cloud infrastructure, while the *Access Control Engine* was running on *avalon.isti.cnr.it*. The probes that generate events related to the sensors and the actuators are running on top of the middleware: *energia.isti.cnr.it* and the events are flowing through the message broker *AMQ* running on *atlantis.isti.cnr.it*.

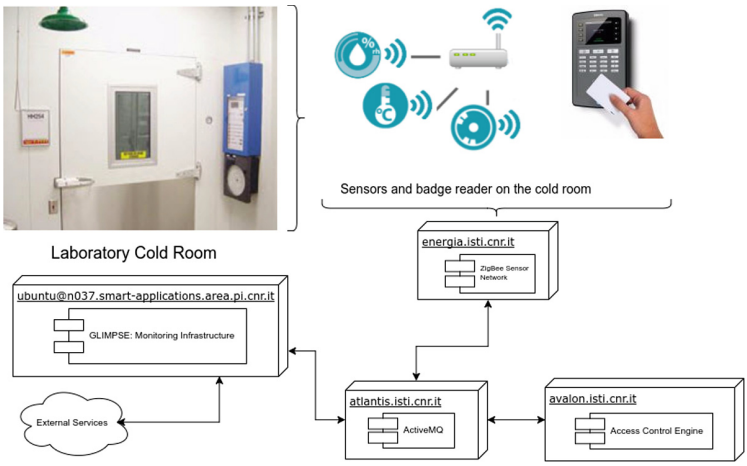

Fig. 4. Deployment configuration

5.2 Sensors, Access and Usage Rules Management

In this Section, we focus on the interaction between *CEP - Events*, *CEP - Usage* and the Access control Engine for the enforcement of the sensors, access and

usage rules. Specifically, as shown in Fig. 1, through the *Rules Editor* the Administrators loaded the sensors rule useful for monitoring the sensors status and the access rules for the identification of who is currently asking resource access.

When an employee, through the RFID Bagde Reader, tries to access the LRC, the *CEP - Events* receives the access request and extracts the room ID and the badge ID. By querying the ISTI-CNR internal employees database, the *CEP - Events* retrieves (Role, Owner room id) attributes related to the user who is asking for the access. Using the data collected, the *CEP - Events* sends a *Policy evaluation request* through the *Rest Engine* to the *Access Control Engine* node and a *PdpAccessRequest* event to the *CEP - Usage* to notify that an access request has been sent. The *PEP* translates the request into an XACML access request and sends it to the *PDP*, which evaluates the request according to the access control policies injected by *PAP*, and sends back the response to the*PEP*, which in turn sends back to the *CEP - Usage* through the *Rest Engine*.

```
1   [..setup and import omitted..]
2
3   declare SensorFailureEvent
4    @idroom: int
5    @idsensor: int
6   end
7
8   rule "Check␣data␣from␣temperature␣sensor"
9     no-loop true
10    salience 1
11    dialect "java"
12  when
13    $aEvent:GlimpseBaseEventSB(this.isConsumed == false, this.isException == false,
14      (this.getTemperature == null || < -20 || > 0 ) );
15
16    $bEvent:GlimpseBaseEventSB(
17      this.isConsumed == false, this.isException == false,
18      (this.getTemperature == null || < -20 || > 0 ),
19      this after $aEvent, this.getSensorID == $aEvent.getSensorID);
20  then
21    SensorFailureEvent failureDetected = new SensorFailureEvent(idRoom,idSensor);
22    CepBinder.sendEventTo("CEP␣-␣Usage", failureDetected);
23    $aEvent.setConsumed(true);  $bEvent.setConsumed(true);
24    retract($aEvent);  retract($bEvent);
25  end
```

Listing 1. Sensors rule

An example of a sensors rule used by the *CEP - Events* for controlling all the installed sensors is shown in Listing 1. In particular, when the *CEP - Events* receives from the monitored sensors, for two consecutive times, null or out-of-range values (lines 13–14 and 17–18 of Listing 1), the *CEP - Events* generates a complex event called *SensorFailureEvent* for notifying the detected failure to the *CEP - Usage*, so that it can activate the corrective actions. For confidential reasons, we do not provide here the complete specification of the XACML access control policies adopted for managing access to the different rooms inside the ISTI-CNR research area. As reported in Listing 2, we just show an extract of some of the rules implemented in ISTI-CNR access control policies so to better explain the potentialities and features of the proposed infrastructure. Among the different types of rooms (resources) of the ISTI-CNR access control policies, here we focus on three of them: *common room, office room*, and *LCR*.

The ISTI-CNR access control policies specify different kinds of employees (subjects); however, considering the LRC, the most important are: *Biologist, Physician* and *Technician*. The policies manage also several types of actions for each room, however in this section we only consider the simpler one: the *access*.

Finally, the ISTI-CNR access control policies specify different environment values and conditions; for aim of simplicity here we only consider the case in which the environment represents the different time slots, in which an employee can access the different rooms. Considering Listing 2. the rules specify that:

1. Rule 1: each employee can access his own office and the common rooms during the business-time (from 8am to 8pm);
2. Rule 2: only an employee, who is either *Biologist* of *Physician*, can access the *LCR* during the business-time;
3. Rule 3: The *Technician* can access the *LCR* at any time.

```
<Policy PolicyId="SmartPolicy" RuleCombiningAlgId="first-applicable">
  <Target>
   <Subjects><Subject><SubjectMatch MatchId="string-equal">
      <AttributeValue DataType="string">employee</AttributeValue>
     </SubjectMatch></Subject></Subjects>
   <Resources><Resource><ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">CNR</AttributeValue>
     </ResourceMatch></Resource></Resources>
   <Actions><Action><ActionMatch MatchId="string-equal">
      <AttributeValue DataType="string">acces</AttributeValue>
     </ActionMatch></Action></Actions>
  </Target>
  <Rule RuleId="Rule1" Effect="Permit">
   <Target>
   <Resources>
    <Resource><ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">CNR</AttributeValue>
     </ResourceMatch>
     <ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">office room</AttributeValue>
     </ResourceMatch></Resource>
    <Resource><ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">CNR</AttributeValue>
     </ResourceMatch>
     <ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">common room</AttributeValue>
     </ResourceMatch></Resource>
   </Resources>
   <Environments>
    <Environment><EnvironmentMatch MatchId="time-equal">
      <AttributeValue DataType="time">8:00:00</AttributeValue>
     </EnvironmentMatch></Environment>
    <Environment><EnvironmentMatch MatchId="dayTimeDuration-equal">
      <AttributeValue DataType="dayTimeDuration">PT12H</AttributeValue>
    </EnvironmentMatch></Environment></Environments>
  </Target>
 </Rule>
 <Rule RuleId="Rule2" Effect="Permit">
  <Target>
    <Resources><Resource><ResourceMatch MatchId="string-equal">
       <AttributeValue DataType="string">LCR</AttributeValue>
      </ResourceMatch></Resource></Resources>
    <Environments>
     <Environment><EnvironmentMatch MatchId="time-equal">
       <AttributeValue DataType="time">8:00:00</AttributeValue>
      </EnvironmentMatch></Environment>
     <Environment><EnvironmentMatch MatchId="dayTimeDuration-equal">
       <AttributeValue DataType="dayTimeDuration">PT12H</AttributeValue>
       </EnvironmentMatch></Environment></Environments>
   </Target>
   <Condition><Apply FunctionId="string-at-least-one-member-of">
     <AttributeValue DataType="string">biologist</AttributeValue>
     <AttributeValue DataType="string">physician</AttributeValue>
    </Apply></Condition>
  </Rule>
  <Rule RuleId="Rule3" Effect="Permit">
   <Target>
    <Subjects><Subject><SubjectMatch MatchId="string-equal">
      <AttributeValue DataType="string">technician</AttributeValue>
    </SubjectMatch></Subject></Subjects>
    <Resources><Resource><ResourceMatch MatchId="string-equal">
      <AttributeValue DataType="string">LCR</AttributeValue>
    </ResourceMatch></Resource></Resources>
   </Target>
  </Rule>
  <Rule RuleId="default" Effect="Deny"/>
</Policy>
```

Listing 2. Smart environment access policy

At run time, the request sent by the *CEP - Events* to the *Access Control Engine* is evaluated by the PDP component and the corresponding reply is sent back to the *CEP - Usage*, which uses the received (permit or deny) response to allow the resource access or to deny it in case of possible violations or resource misuses. In both cases, the *CEP - Usage* is in charge of notifying the Actuators of the (corrective) actions to be executed. An example of usage rule implemented by the *CEP - Usage* is shown in Listing 1. In particular, the rule checks if there are pending access requests to the *LCR* and ongoing alarms. In this last case, it retrieves from the employee data base the contact data of the technician in charge of managing the alarm and it inhibits any possible access to *LCR*, apart from the selected technician.

5.3 Maintenance Activity Scenario

This section describes the management of the scenario in which an alarm is raised by the sensors and a corrective maintenance request is sent to the technician. The scenario preconditions are the following: (i) Each employee is registered on the internal ISTI-CNR personal data base; (ii) Each employee accesses the different rooms by means of a personal badge equipped with a beacon bluetoot, as shown in Fig. 2; (iii) The *LCR* room is closed by default and constantly monitored by sensors able to send events to the Monitoring Infrastructure; (iv) No one is currently inside the LCR and sensor values are within their allowed ranges.

Initially, a *Physician* requires to access the LCR by using the LCR RFID badge reader connected the to nearest network. According to the interaction described in Sect. 5.2, an event is sent to the *CEP - Events* through the *Message Broker* and the proper access request is sent to the *Access Control Engine*. This last evaluates the request and sends back the response to the *PEP*, which in turn sends back to the *CEP - Usage* through the *REST Engine*.

As shown in Listing 3, if: any revocation of permission is ongoing (line 8), there are not critical conditions (i.e. the values of temperature, humidity, energy consumption, noise are in the allowed ranges - line 18), and PDP response includes a permit (i.e. *Physician* requires to access the LCR during the business-time -line 15), the *CEP - Usage* sends an event to the *Actuator gateway* for enabling the door opening through the *Response Dispatcher*.

Supposing instead that a critical condition has been detected by *CEP - Events*, for instance either the power consumption sensors is out of range or there are significant variations in the noise or the temperature is in a not allowed range, a *SensorFailureEvent* event is sent to the *CEP - Usage* (line 46 of Listing 1). This last overrides the PDP response allowing the access to the technician only and sends an event to the *Actuator gateway* for enabling the door opening to the technician only (line 50–51 of Listing 3).

Moreover, the *CEP - Usage* sends an alarm event to the Supervision through a specific *Actuator gateway* for requesting exceptional maintenance of the LCR (line 37 of Listing 3).

```
[..setup and import omitted..]

declare SensorFailureEvent
 @idroom: int
 @idsensor: int
end

rule "If there are NOT pending alarm forward PDP access response"
no-loop true
salience 1
dialect "java"

when
  $aEvent:PdpAccessRequest();
  $bEvent:PdpAccessResponse(this.isConsumed == false, this.isException == false,
    this.getIdRequest == $aEvent.getIdRequest, ($bEvent.getResponse == "Permit" || "Deny"),
    this after $aEvent);
  not(SensorFailureEvent(this.isConsumed == false, this.isException == false,
    this.idRoom == $aEvent.idRoom, this.idSensor == $aEvent.idSensor));

then
  Actuators.ManageAccess($aEvent.getIdSensor(),$aEvent.getIdroom(), $bEvent.getResponse()));
end

rule "If there are failures take countermeasures"
no-loop true
salience 1
dialect "java"

when
  $aEvent:SensorFailureEvent();
then
  Alarm.NotifyToSupervision($aEvent.idsensor, $aEvent.idroom);
end

rule "If there are pending alarm check accesses"
no-loop true
salience 1
dialect "java"

when
  $aEvent:PdpAccessRequest();
  $bEvent:PdpAccessResponse(this.isConsumed == false, this.isException == false,
    this.getIdRequest == $aEvent.getIdRequest, ($bEvent.getResponse == "Permit" || "Deny"),
    this after $aEvent);
  $cEvent:SensorFailureEvent(this.isConsumed == false, this.isException == false,
    this.idRoom == $aEvent.idRoom);

then
  Actuators.ManageAccess($aEvent.getIdSensor(),$aEvent.getIdroom(),
            PersonnelDatabase.checkIfIsTechnician($aEvent.getIdUser));
end
```

Listing 3. Usage rule

5.4 Results Analysis and Lesson Learned

For space limitation we just provided very few details about the adoption of the proposed infrastructure inside ISTI-CNR research area. The experiment described here is part of a larger one that will involve control of all the ISTI-CNR area. The main peculiarity of the proposed approach is that the control of the different rooms is not centralized, but can be specialized time to time according the different research exigencies and in agreement with general administrative and security regulations.

Though the proposed infrastructure each lab head, or even each researcher, has the freedom to specify its own control rules depending on the sensors installed in the room or the required behavior, without changing those for the rest the area. Considering specifically the LRC, the infrastructure proposed let a detailed control management of the many PhD students requiring the access to the laboratory. Indeed without a deep impact on the generic control rules, and again in agreement with administrative and security regulations, it was possible to

differentiate for each PhD student, both the allowed access time and the allowed activity when specific experimentation were ongoing.

From this experimentation two main considerations have come to light:

- Different stakeholders may have different views of the control management of a specific environment, sometimes even ignoring which are the common best practices or the corrective activities. In the specific case of LRC defining precisely the Sensors, Usage and Access control rules required many interviews and interactions with different ISTI personnel having separate competencies: researchers from one side and technicians from the other. However, the adoption of the proposed infrastructure forced them to provide, for the first time, the documentation of the procedures ruling the LRC laboratory, to highlight the critical points both from the sensors and usage point of view and to define precisely responsibilities and activities to be performed in case of security and safety flaws.
- Leaving the freedom of each lab head to define the more suitable control rules, evidenced a stringent necessity to improve the infrastructure with more dynamic features for a careful validation of consistency and correctness of the set of rules so to avoid violations of the general administrative and security regulations.

6 Related Work

This work spans over several research directions, including: smart environment, access and usage control and monitoring approaches.

Enabling Platforms in Smart Environments: The SE paradigm depends on communication and cooperation between numerous devices, sensor networks embedded in the environment itself, servers in a fixed infrastructure and the increasing number of mobile devices carried by people. In order to enable the SE paradigm, diverse platforms and software infrastructures have been proposed in the literature [29]. Among these, FI-WARE[8] is emerging as a core standard platform for Smart and Connected Communities (SCC) [30,31]. The FI-WARE project is producing new tools to facilitate the development of application and fostering a major inclusion of software standards for smart cities [32]. These tools are provided as Generic Enablers (GE): software components that can be configured and deployed on a cloud platform in order to easily implement an application. Another important enabling platform for SE is represented by the universAAL architecture[9], with a particular focus on IoT and Ambient Assisted Living (AAL) [33]. Besides its concrete open source implementation, universAAL proposes an architectural reference model based on a set of virtual communication buses as semantic brokers for context events, semantic service requests (and

[8] http://www.fiware.org.
[9] http://www.universaal.info/.

responses), and user interaction functions. For these reasons, we used the separation of concerns and the service discovery capabilities offered by the universAAL reference model to build the middleware architecture proposed in this paper.

Access and Usage Control: Concerning the testing of access control policies, combinatorial approaches have been proven to be effective in the automated generation of the test cases (access requests). Among the various proposals, the Targen tool [34] generates test inputs by using combinatorial coverage of the truth values of independent clauses of XACML policy values, while the X-CREATE tool [35] relies on combinatorial approaches of the subject, resource, action and environment values taken from the XACML policy. The main advantage of this approach with respect to Targen is the higher structural variability of the derived test inputs able to guarantee the coverage of the input domain of the XACML policy. Other works address model-based testing and provide a methodology for the generation of test cases based on combinatorial approaches of the elements of the model (role names, permission names, context names).

Concerning the Usage Control systems, the work in [36] proposes a usage control model based on UCON and describes a framework to implement it in an operating system kernel, on top of the existing DAC mechanism. Other available solutions, such as [37], propose proactive mechanisms for preventing possible policy violations and present a combination of runtime monitoring and self-adaptation to simplify the autonomic management of authorization infrastructures. In recent years, as surveyed in [38], testing of authorization systems has been focused on evidencing the appropriateness of the UCON enforcement mechanism, focusing on the performance analysis or establishing proper enforcement mechanisms by means of formal models. Finally, the authors of [39] address the testing of the Policy Decision Point (PDP) implementation within the PolPA authorization system, which enables history-based and usage-based control of accesses proposing two testing strategies specifically conceived for validating the history-based access control and the usage control functionalities of the PolPA PDP.

Monitoring: Several general-purpose monitoring proposals are currently available, which can be mainly divided into two groups: those that are embedded in the execution engine, such as [40,41], and those that can be integrated into the execution framework as an additional component, such for instance [42–44]. Both the solutions have specific advantages. For sure, an embedded solution reduces the performance delay of the execution framework, mainly in terms of interactions and communication time. Coverage indicators can be directly evaluated by the execution framework, which can also execute corrective actions in case of important deviations. The main disadvantage of these approaches is the lack of flexibility in the data collection, the coverage measure definition and the language adopted.

7 Conclusions

In this paper, we proposed an infrastructure for runtime management and control of smart environments. The main advantages of the proposed solution are its

flexibility and the possibility of decoupling the different levels of rules that are defined and implemented for managing the resources of the smart environments and regulate the access to them. Specifically, three levels of rules are defined: the *Sensor Rules* for correlating sensors data and technologies; the *Usage Control Rules*, which define the users and sensors interactions; and the *Access Control Rules*, which manage the accesses to the different resources expressed through a specific control policy formalism. This allows an easy maintenance and updating of control rules when context changes or constraints violations take place.

A first validation on a real cased study, considering a medical cold storage and implementing an XACML policy, has been described. The presented scenario evidenced the effectiveness of the proposed approach to correlate events generated by different sensors and to leverage different levels of rules for raising alarms, when critical situations are detected.

As a future work, we would like to validate the proposed solution in other smart environments with different peculiarities and security constraints as well as different access control policy specification languages. Moreover, we plan to extend the infrastructure to include more refined levels of rules, further decoupling the management and control functionalities of the proposed infrastructure.

Acknowledgments. This work has been partially funded by the projects GAUSS, a National Research project (MIUR, PRIN 2015, Contract 2015KWREMX), and SIGS (FAR/FAS 2014-2016 research program of the Tuscany Region).

References

1. Weiser, M.: The computer for the 21st century. Sci. Am. **265**(3), 94–104 (1991)
2. Tragos, E.Z., Bernabe, J.B., Staudemeyer, R.C., Luis, J., Ramos, H., Fragkiadakis, A., Skarmeta, A., Nati, M., Gluhak, A.: Trusted IoT in the complex landscape of governance, security, privacy, availability and safety. In: Digitising the Industry-Internet of Things Connecting the Physical, Digital and Virtual Worlds. River Publishers Series in Communications, pp. 210–239 (2016)
3. Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley, Hoboken (2010)
4. Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti, E.: An automated testing framework of model-driven tools for XACML policy specification. In: 9th QUATIC 2014, Guimaraes, Portugal, 23–26 September 2014, pp. 75–84 (2014)
5. Daoudagh, S., Kateb, D.E., Lonetti, F., Marchetti, E., Mouelhi, T.: A toolchain for model-based design and testing of access control systems. In: MODELSWARD 2015 Angers, Loire Valley, France, 9–11 February 2015, pp. 411–418 (2015)
6. Ahmad, A., Rathore, M.M., Paul, A., Hong, W.H., Seo, H.: Context-aware mobile sensors for sensing discrete events in smart environment. J. Sens. (2016)
7. Drăgoicea, M., Bucur, L., Pătraşcu, M.: A service oriented simulation architecture for intelligent building management. In: Falcão e Cunha, J., Snene, M., Nóvoa, H. (eds.) IESS 2013. LNBIP, vol. 143, pp. 14–28. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36356-6_2
8. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. Computer **29**(2), 38–47 (1996)

9. OASIS Standard: eXtensible Access Control Markup Language (XACML) Version 2.0 (2005)
10. Park, J., Sandhu, R.: The UCON ABC usage control model. ACM Trans. Inf. Syst. Secur. (TISSEC) **7**(1), 128–174 (2004)
11. Calabrò, A., Lonetti, F., Marchetti, E.: Monitoring of business process execution based on performance indicators. In: 41st, EUROMICRO-SEAA 2015, Madeira, Portugal, 26–28 August 2015, pp. 255–258 (2015)
12. Calabrò, A., Lonetti, F., Marchetti, E.: KPI evaluation of the business process execution through event monitoring activity. In: ES 2015, Basel, Switzerland, 14–15 October 2015, pp. 169–176 (2015)
13. Czarnecki, K., Eisenecker, U.W.: Generative Programming - Methods, Tools and Applications. Addison-Wesley, Boston (2000)
14. Bertolino, A., Calabrò, A., De Angelis, G.: A generative approach for the adaptive monitoring of SLA in service choreographies. In: Daniel, F., Dolog, P., Li, Q. (eds.) ICWE 2013. LNCS, vol. 7977, pp. 408–415. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39200-9_34
15. Drools, J.: Drools Fusion: Complex Event Processor. http://www.jboss.org/drools/drools-fusion.html
16. Wilde, E., Pautasso, C. (eds.): REST: From Research to Practice. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-8303-9
17. Apache: Apache ActiveMQ. http://activemq.apache.org/
18. Oracle: Java message service documentation. Technical report (2016). http://www.oracle.com/technetwork/java/docs-136352.html
19. Barsocchi, P., Oligeri, G., Potortì, F.: Measurement-based frame error model for simulating outdoor Wi-Fi networks. IEEE Trans. Wirel. Commun. **8**(3), 1154–1158 (2009)
20. Furfari, F., Girolami, M., Lenzi, S., Chessa, S.: A service-oriented zigbee gateway for smart environments. J. Ambient Intell. Smart Environ. **6**(6), 691–705 (2014)
21. Palumbo, F., Ullberg, J., Štimec, A., Furfari, F., Karlsson, L., Coradeschi, S.: Sensor network infrastructure for a home care monitoring system. Sensors **14**(3), 3833–3860 (2014)
22. Barsocchi, P., Ferro, E., Fortunati, L., Mavilia, F., Palumbo, F.: EMS@CNR: an energy monitoring sensor network infrastructure for in-building location-based services. In: 2014 International Conference on High Performance Computing & Simulation (HPCS), pp. 857–862. IEEE (2014)
23. Barbon, G., Margolis, M., Palumbo, F., Raimondi, F., Weldin, N.: Taking arduino to the internet of things: the ASIP programming model. Comput. Commun. **89**, 128–140 (2016)
24. Palumbo, F., La Rosa, D., Chessa, S.: GP-m: mobile middleware infrastructure for ambient assisted living. In: 2014 IEEE Symposium on Computers and Communication (ISCC), pp. 1–6. IEEE (2014)
25. Kim, Y., Schmid, T., Charbiwala, Z.M., Srivastava, M.B.: ViridiScope: design and implementation of a fine grained power monitoring system for homes. In: Proceedings of the 11th International Conference on Ubiquitous Computing, pp. 245–254. ACM (2009)
26. Girolami, M., Palumbo, F., Furfari, F., Chessa, S.: The integration of ZigBee with the GiraffPlus robotic framework. In: O'Grady, M.J., Vahdat-Nejad, H., Wolf, K.-H., Dragone, M., Ye, J., Röcker, C., O'Hare, G. (eds.) AmI 2013. CCIS, vol. 413, pp. 86–101. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04406-4_10

27. Palumbo, F., Barsocchi, P., Furfari, F., Ferro, E.: AAL middleware infrastructure for green bed activity monitoring. J. Sens. **2013** (2013)
28. CNR: Smartcampus technical report. Technical report (2016). http://www.smart-applications.area.pi.cnr.it/doc/smartareaTechnicalDescription.pdf
29. Namiot, D., Sneps-Sneppe, M.: On software standards for smart cities: API or DPI. In: Proceedings of the 2014 ITU Kaleidoscope Academic Conference: Living in a Converged World-Impossible Without Standards?, pp. 169–174. IEEE (2014)
30. Glikson, A.: Fi-ware: core platform for future internet applications. In: Proceedings of the 4th Annual International Conference on Systems and Storage (2011)
31. Sun, Y., Song, H., Jara, A.J., Bie, R.: Internet of things and big data analytics for smart and connected communities. IEEE Access **4**, 766–773 (2016)
32. Ramparany, F., Marquez, F.G., Soriano, J., Elsaleh, T.: Handling smart environment devices, data and services at the semantic level with the FI-ware core platform. In: Big Data 2014, pp. 14–20. IEEE (2014)
33. Salvi, D., Montalva Colomer, J.B., Arredondo, M.T., Prazak-Aram, B., Mayer, C.: A framework for evaluating ambient assisted living technologies and the experience of the universaal project. J. Ambient Intell. Smart Environ. **7**(3), 329–352 (2015)
34. Martin, E., Xie, T.: Automated test generation for access control policies. In: Supplemental Proceedings of 17th International Symposium on Software Reliability Engineering (ISSRE), November 2006
35. Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti, E., Schilders, L.: Automated testing of extensible access control markup language-based access control systems. IET Softw. **7**(4), 203–212 (2013)
36. Teigao, R., Maziero, C., Santin, A.: Applying a usage control model in an operating system kernel. J. Netw. Comput. Appl. **34**(4), 1342–1352 (2011)
37. Bailey, C.: Application of self-adaptive techniques to federated authorization models. In: 2012 34th International Conference on Software Engineering (ICSE), pp. 1495–1498. IEEE (2012)
38. Nyre, Å.A.: Usage control enforcement - a survey. In: Tjoa, A.M., Quirchmayr, G., You, I., Xu, L. (eds.) CD-ARES 2011. LNCS, vol. 6908, pp. 38–49. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23300-5_4
39. Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti, E., Martinelli, F., Mori, P.: Testing of PolPA-based usage control systems. Softw. Q. J. **22**(2), 241–271 (2014)
40. Daoudagh, S., Lonetti, F., Marchetti, E.: Assessment of access control systems using mutation testing. In: TELERISE 2015, Florence, Italy, 18 May 2015, pp. 8–13 (2015)
41. Bertolino, A., Daoudagh, S., Kateb, D.E., Henard, C., Traon, Y.L., Lonetti, F., Marchetti, E., Mouelhi, T., Papadakis, M.: Similarity testing for access control. Inf. Softw. Technol. **58**, 355–372 (2015)
42. Carvallo, P., Cavalli, A.R., Mallouli, W., Rios, E.: Multi-cloud applications security monitoring. In: Au, M.H.A., Castiglione, A., Choo, K.-K.R., Palmieri, F., Li, K.-C. (eds.) GPC 2017. LNCS, vol. 10232, pp. 748–758. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57186-7_54
43. Bertolino, A., Calabró, A., Lonetti, F., Marchetti, E.: Towards business process execution adequacy criteria. In: Winkler, D., Biffl, S., Bergsmann, J. (eds.) SWQD 2016. LNBIP, vol. 238, pp. 37–48. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27033-3_3
44. Calabrò, A., Lonetti, F., Marchetti, E., Spagnolo, G.O.: Enhancing business process performance analysis through coverage-based monitoring. In: 10th QUATIC 2016, Lisbon, Portugal, 6–9 September 2016, pp. 35–43 (2016)