J Med Syst (2016) 40: 238
DOI 10.1007/s10916-016-0581-7

SYSTEMS-LEVEL QUALITY IMPROVEMENT

Semantically Enriched Data Access Policies in eHealth

Michał Drozdowicz[1] · **Maria Ganzha**[2] · **Marcin Paprzycki**[1]

Received: 20 November 2015 / Accepted: 31 August 2016 / Published online: 24 September 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Internet of Things (IoT) requires novel solutions to facilitate autonomous, though controlled, resource access. Access policies have to facilitate interactions between heterogeneous entities (devices and humans). Here, we focus our attention on access control in eHealth. We propose an approach based on enriching policies, based on well-known and widely-used eXtensible Access Control Markup Language, with semantics. In the paper we describe an implementation of a Policy Information Point integrated with the *HL7 Security and Privacy Ontology*.

Keywords IoT · XACML · Ontologies · Semantic technologies · Access management · eHealth · HL7

Introduction

Rising number of connected devices, including networks of sensors, opens questions related to data privacy and security; on all levels of the ecosystem. Regulation of access to data and services, exposed by the "components" of the IoT, is similar to that of the Web resources and services. There is an "entity", described with attributes, or roles that requests access to data and/or resource(s) (or services available within resources). In response, based on declarative or imperative rules, such request is granted (or denied). In the IoT, the core use-case involves interacting devices (rather than human-computer interactions). Furthermore, simple attribute or role-based approaches to access control, may not work. The main reasons are: (i) huge number of resources ([2, 12]), (ii) fast growing number of consumers ([13, 15, 24]), (iii) heterogeneity of data and services ([10, 11, 14, 16]), (iv) dynamics of (often short-lived) interactions ([18, 19, 23]).

Data privacy, and thus access control, is particularly important in eHealth. Here, the Internet of Things brings about a number of potential benefits, such as constant monitoring of patient's medical parameters, online consultations, or remote control of medical devices (e.g. insulin pumps). It also increases the need for access control as, in the IoT scenarios, available patient data becomes even more detailed, making it extremely sensitive. Furthermore, personal data, of a single patient, may be geographically distributed (e.g. across patient's home, multiple medical institutions and devices, etc.).

Here, we discuss a semantically enriched access control policy system. In comparison to our previous publications on the topic, here we present how the solution has been integrated with an existing eHealth privacy ontology and provide a relevant example of benefits that such an approach provides.

To this effect, we start by introducing the state-of-the-art, in the area of interest, in "State-of-the-art". In "Privacy and access control in eHealth", we describe the background

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Marcin Paprzycki
marcin.paprzycki@ibspan.waw.pl

Michał Drozdowicz
michal.drozdowicz@ibspan.waw.pl

Maria Ganzha
m.ganzha@mini.pw.edu.pl

[1] Systems Research Institute Polish Academy of Sciences, Warsaw, Poland

[2] Department of Mathematics and Information Sciences, Technical University of Warsaw, Warsaw, Poland

for employing policy-based access control in the field of eHealth, as well as the unique challenges arising when it is combined with the Internet of Things technologies. "XACML", provides a brief introduction to the XACML policy language that we have selected as the basis of our work. We follow, in "HL7 security and privacy ontology", with an introduction to the *HL7 Security and Privacy Ontology*. Having provided the details of the work we have built upon, in "Semantic policy information point", we describe the architecture of the Semantic Policy Information Point. "Example of enforcing access control" gives an example of using the semantic extensions to the XACML, while evaluating a data access request against a policy defined in the system. We follow with an outline of the proposed approach to the (re)design of the *Policy Information Point*.

State-of-the-art

Initial study of pertinent literature has been included in the conference version of this paper. There, we have discussed state-of-the-art concerning: (a) policy based access control, (b) semantic approaches to access control policies, and (c) semantic extensions to the XACML (for a complete presentation see, [22]). Therefore, here we focus our attention on the remaining / newly found reference-points that jointly provide context for our present work.

There have been attempts at implementing policy based access control in the context of access control in the domain of eHealth and pervasive healthcare. In [27], a framework for context-aware Role Based Access Control, that utilizes XACML for policy specification and enforcement together with an ontology is proposed. In this solution, a semantic reasoner is used for dynamic role assignment, depending on the access control request context (both domain-specific and environmental). With respect to this research, our approach makes broader use of semantic inference – providing not only additional subject role specification, but also enriching the general attribute space of the decision context, thus focusing more on the Attribute Based Access Control model.

In [26], a similar approach is suggested, bringing more attention to the problem of heterogeneity and interoperability between systems using different vocabularies. While the solution provides great flexibility in defining rules and mapping concepts to a common ontology, in our opinion, its reliance on a custom policy engine results in limited robustness and practical applicability.

One important finding from our discussions with colleagues outside of the academia is that, while standards slowly find their way to EHR systems, providing greater uniformity across solutions and easier integration between institutions, access control is typically handled in a proprietary way. In this context, our proposal to build on top of the XACML as the policy language and standard ontologies, such as the *HL7 Security and Privacy Ontology*, may contribute to seeing wider adoption of more general / interoperable methods.

Privacy and access control in eHealth

It should be obvious that eHealth solutions, placed within the IoT, require better managed data access. Currently, patients' medical records are usually protected only through a coarse-grained access control – based on simple lists of system users, perhaps extended with a role-permission model. However, when looking at the medical information, it is possible to distinguish different levels of data confidentiality and sensitivity. For example, the HL7 standard defines 6 levels of confidentiality [6] and the following sensitivity categories [5]:

- *ETH* – substance abuse information sensitivity
- *GDIS* – genetic disease information sensitivity
- *HIV* – HIV/AIDS information sensitivity
- *PSY* – psychiatry information sensitivity
- *SDV* – sexual assault, abuse, or domestic violence information sensitivity
- *SEX* – sexuality and reproductive health information sensitivity
- *SICKLE* – sickle cell
- *STD* – sexually transmitted disease information sensitivity
- *TBOO* – taboo

On the other hand, note that the information about body parameters (blood pressure, temperature etc.) can (and should) be accessible to multiple "personnel roles", e.g. nurses, physiotherapists, etc.

A typical example, where finer grained access control could provide better privacy, involves a hospital, where (typically) every physician has full access to the medical history of every patient. However, such access is not necessary. It would suffice if the physician had access to the information about her patients, with some additional privileges granted during her "work shifts" (when she would need access to data of "all patients"). Similarly, access to data generated by ("external") devices, gathering information about the patient (or, more importantly, controlling treatment parameters), should be restricted to the explicitly defined personnel.

Another interesting question, concerning access control in "IoT for eHealth", is: where access authorization takes place. Obviously, if all data is curated by a

single institution, access policies are enforced there. With the IoT, however, most sensors are accessible via a gateway (not directly), responsible for securing the connection and, sometimes, also for aggregating and storing the acquired information. Consequently, it becomes possible to provide an additional authorization point within the gateway, which could be a part of the patient's "own infrastructure". Such solution (i) makes data access management more complex for the medical institution, but (ii) gives more control to the owner of the data, and (iii) may be valuable when the patient is treated in multiple institutions (by multiple doctors). Here, access control may be based on patient-doctor relation, rather than associated with specific institutions.

XACML

Keeping this in mind, let us now briefly describe one of most popular approaches to managing data access. The eXtensible Access Control Markup Language (XACML; [4]) is a declarative language for specifying Attribute Based Access Control (ABAC; [25]) policies. It uses the XML as the internal format, but many implementations handle information transfer in other formats, e.g. the SAML ([1]). For a more detailed description of the language and analysis of benefits and drawbacks of using it to specify policy rules we refer to [22]. Here, we will focus on the reference architecture of an XACML processing system, to provide the necessary background needed to understand how our extension, described in further sections, fits into it.

To make the following description of the XACML engine more comprehensible let us assume that we are dealing with a system supporting work of a hospital. The system stores the Electronic Health Record (EHR) data for all its patients. Furthermore it integrates with a number of connected devices, providing remote health monitoring facilities. Obviously, there are many users of the system and rules regulating, which users have access to which parts of patients' data. The XACML provides the possibility to manage and enforce these rules (outside of the services that actually provide the functionality for the users) and enables easy administration and modifications of the policies.

A typical XACML engine consists of the following major components:

- *Policy Enforcement Point* (PEP) – responsible for the actual act of enabling or preventing access to the resource (e.g. to patient's blood pressure record). It also coordinates the execution of, so called, *Obligations*, which are additional operations that should be performed when a decision has been made (e.g. logging the request for auditing purposes).
- *Policy Information Point* (PIP) – a source of attribute values (e.g. specifying that Nick Riviera is a doctor, while Jean Bloom is a patient).
- *Context Handler* – which converts requests and responses between native formats and the XACML canonical representation and coordinates, with the PIPs, gathering of the required attribute values (e.g. checking who is Nick Riviera, asking to access blood pressure data).
- *Policy Decision Point* (PDP) – which evaluates policies and issues the final authorization decisions (e.g. establishes that Nick Riviera as a doctor has the right to access the blood pressure data of Jean Bloom).
- *Policy Administration Point* (PAP) – which defines, stores and manages policies.

Having described the XACML reference architecture, in Fig. 1, we depict the sequence of messages in an access control granting process. Before the system is used, an administrator defines policies within the *PAP* and supplies them to the *PDP*. When a user (e.g. Nick Riviera) wishes to access protected, data (of Jean Bloom), the medical system (Acess Requester) communicates with the *PEP*. The request is forwarded to the *Context Handler*, which in turn notifies the *PDP* of the request. Next, the *PDP* starts evaluating the request against policies, first, retrieving values of all attributes used in policy definitions. These might reflect the identity and role/group information of the user, some properties describing the patient, or the type of the requested EHR information (e.g. finding that Nick Riviera is a doctor, while Jean Bloom is a patient). If the attribute value has been supplied in the request, the *Context Handler* returns it directly. Otherwise, it queries the *PIP*, which can be an EHR store or other data source. When the *PDP* has all the required information, it evaluates the policy rules, combines the results (if multiple policies are in effect) and builds the response context, which is returned to the *Context Handler*. The result of the evaluation are sent to the *PEP*, which enforces the decision (e.g. granting access to the medical data) and may also execute any obligations specified in the policy (e.g. record the event in a security journal).

While a very powerful tool, the XACML lacks possibilities to (i) *reason about the domain* containing the attributes, and (ii) *infer additional data* in an automatic way. Moreover, the XACML standard deals only with the policy definition and enforcement, while it does not provide solutions for the attribute management. To be able to deal with these shortcomings, we propose to introduce semantic data processing. Therefore, let us introduce a pertinent ontology that will allow us to introduce and illustrate the proposed approach.

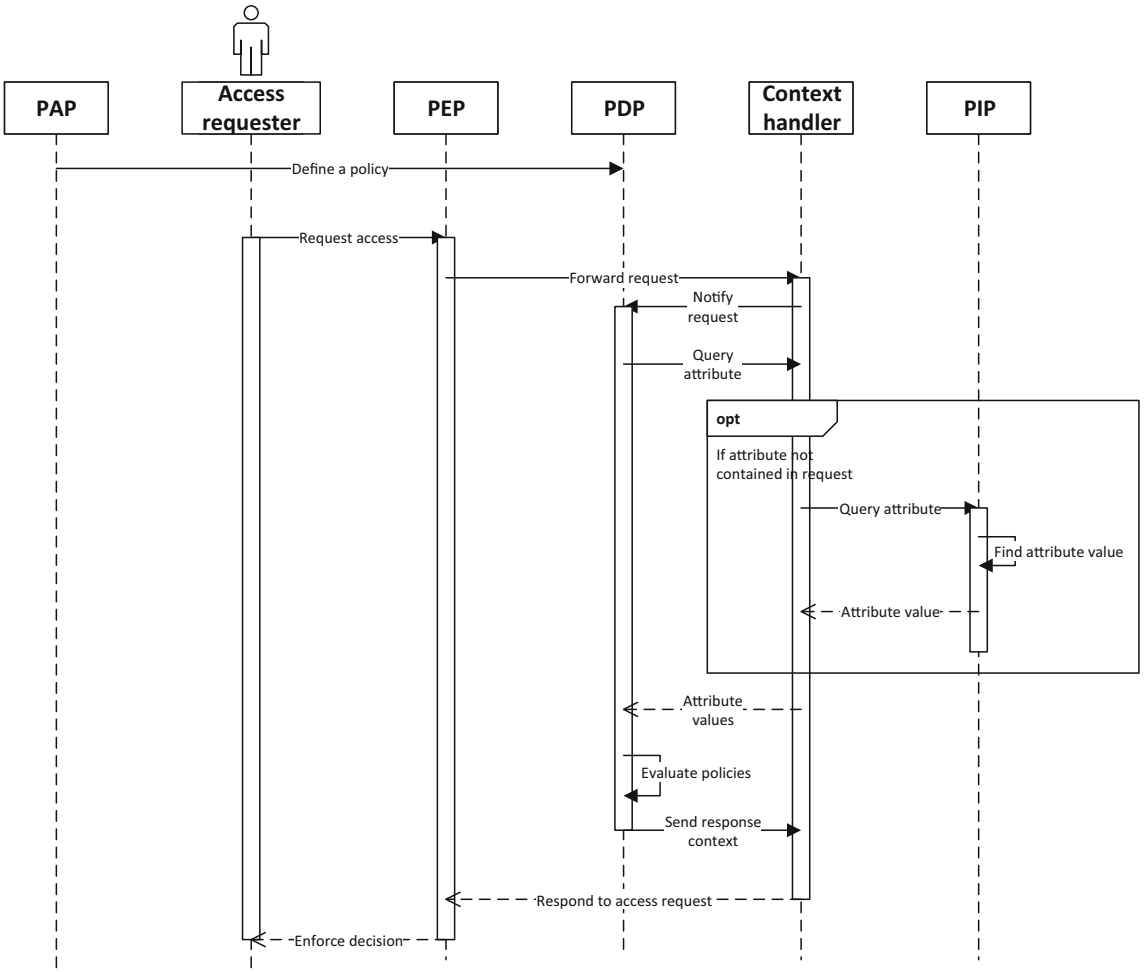

Fig. 1 Evaluation of an access request in a standard XACML architecture

HL7 security and privacy ontology

In [7], we read that *The HL7 Security and Privacy Ontology serves to name, define, formally describe, and interrelate key security and privacy concepts within the scope of Healthcare Information Technology (Healthcare IT)*. This ontology is expressed in OWL, and can be used by itself to provide access control decisions by means of classifying a request using a semantic reasoner. As such, it can be classified as a fully semantic approach to access control policies. However, the ontology can be also used in conjunction with another policy specification language, such as the XACML.

We have found that separating the policy definitions from the domain concepts, reflected in an ontology, provides a more maintainable and expressive solution than defining the policies solely as ontological concepts (as possible through the HL7 ontology). The reasons for that include:

– XACML is widely used in the industry and has very good tool support. As such, it is easier to extend existing XACML systems with a semantic component than it would be to create custom decision points working solely with ontological policies (note that purely semantic approaches have failed in the past, see [22]).
– Combining policies in the HL7 ontology is made by using logical operators allowing to decide whether the request is an instance of a policy or not. In comparison, the XACML standard provides multiple built-in policy combining algorithms and an extension point to define new ones. This solution makes it much easier to express rules such as Deny-overrides or Permit-overrides.
– The datatype restrictions in OWL2 are limited to the XSD facets. The XACML, on the other hand, provides a wider range of functions that can be used for comparing restrictions in the policy with the values of request attributes. It also provides a mechanism for extending the language with custom functions.

Therefore, we have considered if it would be possible to combine the HL7 Security and Privacy Ontology with the XACML-based access control. The remaining parts of this paper discuss how we have actually achieved this goal.

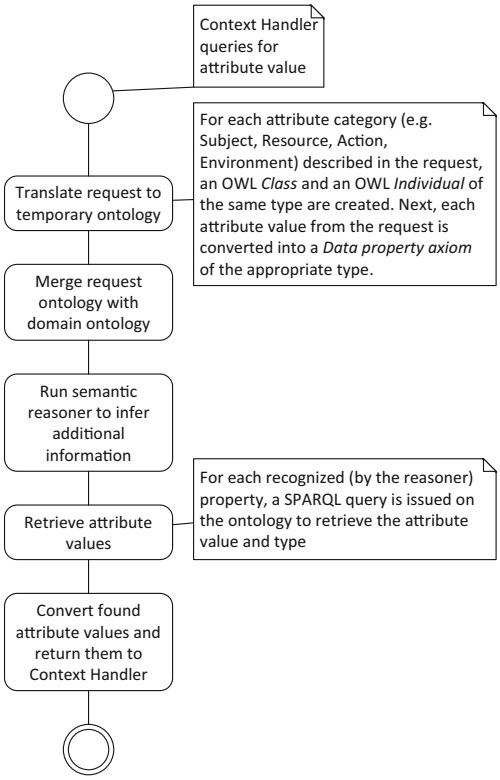

Fig. 2 Algorithm for providing required attribute values

Semantic policy information point

As outlined in [22], following [28], we have proposed a different *Semantic Policy Information Point* (SemanticPIP), for the XACML reference architecture. The proposed SemanticPIP is capable of providing values to unknown attributes, *by inferring them* from the ontologies describing the domain of the system. The general algorithm used by the SemanticPIP is presented in Fig. 2.

Listing 1, shows a sample SPARQL query used for selecting the attribute. Here, the `categoryId` parameter is replaced, during the runtime, with the unique identifier of the *Individual* created in Step 1 – querying Context Handler. For instance, the identifier may represent the user accessing the resource, or the resource under evaluation. Further, the `attributeId` is a fully qualified *id* of the requested attribute (e.g. type of information being accessed).

It should be stressed that, compared with [28], our solution *does not change* the reference architecture of the XACML system. Instead, it implements the contract of the PIP component. Furthermore, the SemanticPIP queries only for, and returns, the attribute values that are requested by the *Context Handler*, reducing the burden placed on the SPARQL engine.

The component has been implemented as an attribute finder extension to the *Balana Framework* ([3, 17]), an XACML engine developed by the WSO2 as a continuation of the popular Sun's XACML Implementation. The engine was chosen due to its maturity and widespread use, among others, as part of the WSO2 Identity Server.

Example of enforcing access control

As an example of the approach, let us consider a scenario in which a blood pressure sensor is working in the patient's (Jean Bloom) apartment. Let us assume that an external system, acting on behalf of a physician (Nick Riviera, authenticated as `nick.riviera@sfhospital.org`, in role `http://hl7.org/ontology/RoleOntology.owl#PhysicianFunctionalRole`), wishes to access the result of the observation (see, Fig. 3). The observation is identified by the *resource-id* of `file://med/bsimpson/20151015/bloodPressure.json` and the *resource-class-id* of `http://drozdowicz.net/sxacml/eHealthSample:BloodPressure`. There exists an XACML policy describing the rules for permitting access. However, as can be seen in Fig. 4, this policy does not cover *specifically* the blood pressure measurement. Instead, it states that a person in a role physician (`http://hl7.org/ontology/RoleOntology.owl#PhysicianFunctionalRole`) may access clinical observations of a patient (identified as *resource-class-id* of `http://hl7.org/ontology/ObjectOntology.owl#ExternalClinicalInformation`).

Apart from the policy, the authorization system also uses an ontology that describes relationships between needed concepts (see, Fig. 5). This ontology is based on the HL7 Privacy and Security Ontology, by using the Object class as an equivalent of the XACML resource.

Acting on the request, the XACML engine asks the SemanticPIP for the `resource-class-id` attribute value. The SemanticPIP uses the semantic reasoner to infer that the BloodPressure class is a subclass of the ExternalClinicalInformation (from the HL7 Security and Privacy ontology). This information is added to the request context and used by the XACML engine to permit the request.

Listing 1 SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?val WHERE
{
 <categoryId> <attributeId> ?val
}
```

Fig. 3 Example XACML request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
    http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd" ReturnPolicyIdList="false">
  <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id">
      <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
        nick.riviera@sfhospital.org </AttributeValue>
    </Attribute>
    <Attribute IncludeInResult="false" AttributeId="net:drozdowicz:sxacml:subject:subject-role-id">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
        http://hl7.org/ontology/RoleOntology.owl#PharmacistFunctionalRole </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
        file://med/bsimpson/20151015/bloodPressure.json </AttributeValue>
    </Attribute>
    <Attribute IncludeInResult="false" AttributeId="sxacml:resource:resource-class-id">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
        http://drozdowicz.net/sxacml/eHealthSample:BloodPressure </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
    <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

Fig. 4 Example XACML policy

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
  xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
  http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd"
  PolicyId="urn:oasis:names:tc:xacml:3.0:example:SimplePolicy1"
  Version="1.0"
  RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides">
  <Description> Springfield Hospital access control policy </Description>
  <Target/>
  <Rule RuleId= "urn:oasis:names:tc:xacml:3.0:example:SimpleRule1" Effect="Permit">
    <Description>
      Any subject with a role of physician can read any health record object.
    </Description>
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
            http://hl7.org/ontology/RoleOntology.owl#PhysicianFunctionalRole
          </AttributeValue>
          <AttributeDesignator MustBePresent="false"
            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-role-id"
            DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
            http://hl7.org/ontology/ObjectOntology.owl#ExternalClinicalInformation
          </AttributeValue>
          <AttributeDesignator MustBePresent="false"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            AttributeId="sxacml:resource:resource-class-id"
            DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
  </Rule>
</Policy>
```

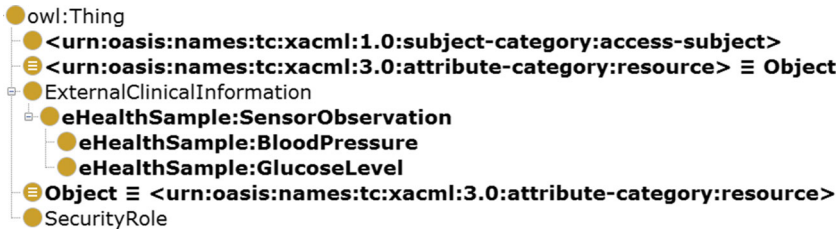

Fig. 5 Ontology enriching the XACML policy

Observe that, in this scenario, the system accessing the data needs not specify that it requests an ExternalClinicalInformation resource – it is not required to know the exact model of the domain on the provider's side. As a matter of fact, using some ontology matching algorithms it would even be possible for the requestor and the owner to communicate using slightly different vocabularies, thus greatly simplifying the interoperation between heterogeneous systems. Moreover, in this example, we have demonstrated the ability to import existing ontologies, namely the RoleOntology and the ObjectOntology, parts of the HL7 Security and Privacy ontology, that already provide concepts describing the domain.

Conclusions and future work

The aim of this paper was to further discuss issues involved in instantiating rule-based resource access policies in the IoT ecosystem. The proposed Semantic Policy Information Point (SemanticPIP) has been implemented and is being thoroughly tested. There are two scenarios where it is going to be applied. First, a non-IoT one, originating from the *Agents in Grid* project (see, [20, 21]). The IoT application, on the other hand, will be focused on users that attend nutritional outpatient care centers. Here a body sensor network will be combined with measurements taking place in the medical facility, leading to a situation where complex data access patterns materialize. This will be realized within the scope of the Inter-IoT EU project. Finally, the SemanticPIP development plans include also support for the additional XACML profiles, namely the Role Based Access Control Profile ([8]) and the Hierarchical Resource Profile ([9]).

Acknowledgments Research of Maria Ganzha and Marcin Paprzycki has been partially supported by the EU-H2020-ICT grant Inter-IoT 687283.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Security assertion markup language (SAML) v2.0. https://www.oasis-open.org/standards, 2005.
2. CASAGRAS final report: RFID and the inclusive model for the Internet of Things. Tech. rep., EU FP7 Project CASAGRAS, 2009.
3. Balana – Open Source XACML 3.0 implementation. http://xacmlinfo.org/2012/08/16/balana-the-open-source-xacml-3-0-implementation/, 2012.
4. eXtensible Access Control Markup Language (XACML) version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, 2013.
5. Hl7 v3 ActInformationSensitivityPolicy value set (oid = 2.16.840.1.113883.1.11.20429). http://www.hl7.org/documentcenter/public_temp_6F42FF1A-1C23-BA17-0C97D0E4BA30D830/standards/vocabulary/vocabulary_tables/infrastructure/vocabulary/ActCode.html#_ActInformationSensitivityPolicy, 2013a.
6. Hl7 v3 confidentiality value set (oid = 2.16.840.1.113883.1.11.10228). http://www.hl7.org/documentcenter/public_temp_6F70C4CC-1C23-BA17-0C20679D971FA269/standards/vocabulary/vocabulary_tables/infrastructure/vocabulary/Confidentiality.html, 2013b.
7. HL7 healthcare privacy and security classification system (HCS). http://www.hl7.org/implement/standards/product_brief.cfm?product_id=345, 2014.
8. XACML v3.0 core and hierarchical role based access control (RBAC) profile version 1.0. http://docs.oasis-open.org/xacml/3.0/rbac/v1.0/xacml-3.0-rbac-v1.0.html, 2014a.
9. XACML v3.0 hierarchical resource profile version 1.0. http://docs.oasis-open.org/xacml/3.0/hierarchical/v1.0/xacml-3.0-hierarchical-v1.0.html, 2014b.
10. Bringing big data to the enterprise. http://www-01.ibm.com/software/data/bigdata/, 2015.
11. Collaborative open market to place objects at your service. http://www.compose-project.eu/, 2015.
12. Internet-of-Things Architecture. http://www.iot-a.eu/public, 2015.
13. IoT@Work. https://www.iot-at-work.eu/, 2015.
14. Open Source cloud solution for the Internet of Things. http://www.openiot.eu/, 2015.
15. uBiquitous, secUre inTernet-of-things with Location and contExawaReness (BUTLER: Smart life. http://www.iot-butler.eu/, 2015.
16. web of Objects ITEA 2 Project. https://itea3.org/project/web-of-objects.html, 2015.
17. WSO2 Balana Implementation. https://github.com/wso2/balana, 2015.
18. Anzelmo, E., Bassi, A., Caprio, D., Dodson, S., Kranenburg, R., and Ratto M: Internet of things. discussion paper, institute for Internet and Society. Berlin, 2011.
19. Atzori, L., Iera, A., and Morabito, G., The internet of things: a survey. In: Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 54, pp. 2787–2805: Elsevier North-Holland Inc., 2010.
20. Drozdowicz, M., Wasielewska, K., Ganzha, M., Paprzycki, M., Attaui, N., Lirkov, I., Olejnik, R., Petcu, D., and Badica, C., Ontology for contract negotiations in agent-based grid resource management system. In: Ivanyi, P., and Topping, B. (Eds.) Trends

in Parallel, Distributed, Grid and Cloud Computing for Engineering. Stirlingshire: Saxe-Coburg Publications, 2011.
21. Drozdowicz, M., Ganzha, M., Wasielewska, K., Paprzycki, M., and Szmeja, P., Using ontologies to manage resources in grid computing: Practical aspects. In: Ossowski, S. (Ed.) Agreement Technologies, Law, Governance and Technology Series, pp. 149–168: Springer, 2013.
22. Drozdowicz, M., Ganzha, M., and Paprzycki, M., Semantic policy information point - preliminary considerations. In: Loshkovska, S., and Koceski, S. (Eds.) ICT Innovations 2015: Engineering Technologies for Better Living, Springer, Advances in Intelligent Systems and Computing, Vol. 399, pp. 11–21, 2016.
23. Evans, D.: The internet of things: How the next evolution of the internet is changing everything. Tech. rep., Cisco Internet Business Solutions Group (IBSG, 2011.
24. FASyS: Absolutely safe and healthy factory. http://www.fasys.es/en/index.php, 2015.
25. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., and Scarfone, K., Guide to attribute based access control (abac) definition and considerations. *NIST Special Publ.* 800:162, 2014.
26. Li, Z., Chu, C. H., and Yao, W., A semantic authorization model for pervasive healthcare. *J. Netw. Comput. Appl.* 38:76–87, 2014.
27. Poulymenopoulou, M., Malamateniou, F., and Vassilacopoulos, G.: An access control framework for pervasive mobile healthcare systems utilizing cloud services. In: Wireless Mobile Communication and Healthcare, Springer, pp 380–385, 2011.
28. Priebe, T., Dobmeier, W., and Kamprath, N.: Supporting attribute-based access control with ontologies. In: The First International Conference on Availability, Reliability, and Security 2006. ARES 2006, p 8. doi:10.1109/ARES.2006.127, 2006.