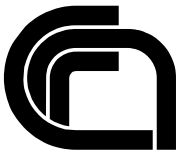


Consiglio Nazionale delle Ricerche

An Expertise-driven Authoring Tool of Privacy Policies for e-Health

R. Conti, I. Matteucci, P. Mori, M. Petrocchi

IIT TR-02/2014

Technical report

Febbraio 2013

Istituto di Informatica e Telematica

An Expertise-driven Authoring Tool of Privacy Policies for e-Health

Riccardo Conti, Ilaria Matteucci, Paolo Mori, Marinella Petrocchi
IIT–CNR, Pisa, Italy
Email:firstname.lastname@iit.cnr.it

***Abstract*—Data sharing on the Internet is crucial in many aspects of nowadays life, from economy to leisure, from public administration to healthcare. However, it implies several privacy issues that have to be managed. Definition of appropriate policies helps to safeguard the data privacy. This paper describes an authoring tool for privacy policies to be applied to the healthcare scenario. The tool exhibits two different interfaces, designed according to specific expertise of the policy authors. It is part of a general framework for editing, analysis, and enforcement of privacy policies. Furthermore, this serves as a first brick for a usability study on such tools.**

I. Introduction

Healthcare organizations provide medical services to their patients, *i.e.,* examinations and diagnostics, and produce electronic documents concerning these services, *e.g.,* reservations, prescriptions, and medical reports. This arises questions concerning, *e.g.,* privacy, confidentiality, and availability of these data. Indeed, the actors of healthcare scenario should be able to access these data, wherever they are stored and whenever they are needed. However, since these documents include sensitive information, their sharing must be regulated by adequate policies to assure the privacy of patients. A privacy policy could state, *e.g.,* that the documents related to a given patient can be accessed by the patient itself, by her general practitioner, and by an emergency doctor in case of accident. These policies can be defined by several entities having the right to restrict the access to these data.

Although some policy languages are currently available (*e.g.,* Ponder, [1], ASL [2] and XACML [3], policy specification is still a difficult task, mainly because of the scarce user-friendliness with these languages. This could prevent individuals from writing complex policies to define fine-grained access rights or to frequently update these policies to fit new needs, and patients from writing policies on their documents at all. Hence, it is pointless to set up an enhanced authorization system adopting a very expressive privacy policy language to protect medical data, if the policy authors are not enabled to edit their policies with a proper authoring tool. The quest for a proper policy authoring tool also follows from the fact that, besides healthcare organization professionals, patients should be enabled to define their access constraints on their documents. In fact, the European Directive 95/46/EC, and its reform IP/12/46 of 25 January 2012, states the right of subjects to define constraints on their personal data.

The goal of this paper is to present a prototype implementation of an authoring tool for defining privacy policies and patient preferences, regulating medical data sharing. The tool has been developed in an *expertise-driven* way, to be tailored for users not familiar with technical policy write up, as well as for more skilled users. It exhibits two interfaces: one addressed to common users, a *Mobile* interface, the other to policy experts, a *Desktop* interface. The idea behind the two interfaces is to provide i) an easy and quick way for a common user (*e.g.,* a patient) to set privacy preferences on her own document in a few click (Mobile interface); ii) the capability to set privacy preferences using a device of common use, such as a smartphone or a tablet (Mobile interface); iii) a mechanism to compose fine-grained privacy policies for skilled users (Desktop interface) that want to set up complex privacy rules. The tool has been developed in accordance to usability guidelines that will be tested in the near future through usability studies that are setting up for the tool assessment.

The paper is structured as follows. Next section presents the policy-based infrastructure that we propose for controlled data access and sharing in e-health. Section III presents the architecture of the authoring tool and focuses on the functionalities of the two user interfaces. Section IV discusses related work on the interdependencies between authoring tools and usability issues. Section V concludes the paper.

II. Policy-based Infrastructure

This section describes the architecture of a policy-based privacy infrastructure, general enough to encompass different use cases in the e-Health privacy management scenario, and supporting the two main phases of a policy lifetime: the *i)* policy generation and the *ii)* policy enforcement. In the first phase, the policy administrators at the healthcare organizations set general privacy policies over the data they host, according to National laws and internal organization planning. Patients as well may express privacy preferences over their medical data, and these preferences are translated in privacy policies. In the second phase, instead, each time a request for accessing a medical data is received, the evaluation of the policies governing access to those data is executed to decide whether the access must be granted or denied. Figure 1 represents the architecture along with the operation workflow.

This work focuses on the policy generation, proposing an authoring tool to enable both healthcare organization professionals and patients to compose their policies and preferences. The policy generation phase consists of the following steps.

1. At system initialisation, the policy experts compose the privacy policies that represent the rules stated by the healthcare organization that produces and stores the data. In some countries, such rules are defined by public agencies and follow

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no 256980 (NESSoS) and from the Registro.it project MobiCare.

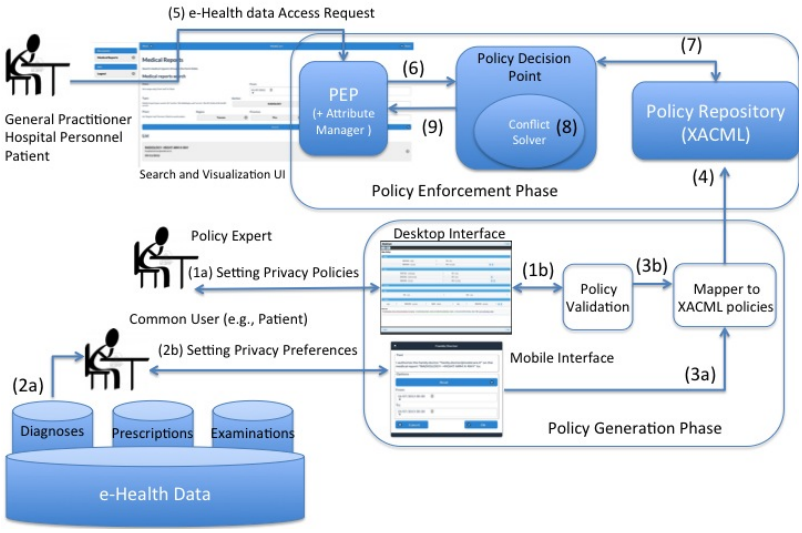


Fig. 1. Policy-based Infrastructure

requirements for protecting sensitive data of private citizens and organizations. The experts use a desktop interface that requires some specific skills in policy specification (1a). In real environments, policy makers set a not negligible number of policies regulating the management and sharing of all the data produced and stored at their healthcare organizations. To come up with a consistent set of policies, policy experts are supported by a validation tool, that guide them in composing conflict-free policies (1b).

2. In the healthcare scenario the subject who produced a document, *e.g.,* a General Practitioner who issued an e-prescription, is entitled to define access restrictions on this document. Moreover, patients should also be able to set privacy preferences on their medical documents (such requirement is defined, for example, by the European Directive for Data Protection 95/46/EC, and its reform IP/12/46 of 25 January 2012). In our model, we assume that, as soon as a new medical document is produced, the patient is notified (2a), and he can set up the privacy preferences on that document through the mobile interface (2b). Each subject can also modify his preferences in a successive step, by querying the medical document repository and choosing which documents will be the object of their privacy preferences. Obviously, patients can express privacy preferences only on their documents and, in general, subjects can only express privacy preferences on the document they have some jurisdiction on.

3. Both the policies written by experts and the privacy preferences expressed by patients and document issuers on their documents are given as input to a mapper that converts them into enforceable policies (3a and 3b). A standard and well supported formalism for enforceable policies is XACML [3].

4. The enforceable policies are stored in the Policy Repository (4) and they will be processed in the policy enforcement phase. For the sake of completeness we also give a brief description of the policy enforcement phase.

5. Different users, such as patients, administrative personnel, doctors, and researchers at the healthcare organizations, try to access some medical documents by formulating an access request through a search and visualization interface (5).

6.The access request is intercepted by a Policy Enforcement Point (PEP) that temporarily suspends the request and invokes a Policy Decision Point (PDP) (6) to evaluate the privacy policies associated to the document whose access is being requested. In our model, we assume that PEP retrieves the attribute values necessary to evaluate the policy (*e.g.,* the requester's credentials, as her identifier and role, and the date, location, and time at which the request has been sent).

7. The PDP retrieves the privacy policies produced in the policy generation phase (7).

8. The PDP evaluates the privacy policies against the access request. In case more than one policy apply to the access request, their results could be in conflict one of each other. A conflict exists when at least two out of a set of policies evaluate a different result (*e.g.,* one policy would allow the request, the other one would deny it). A policy conflict solver is in charge of detecting conflicts in order to solve them (8).

9. The PDP returns the evaluation result to the PEP. Finally, according to the evaluation result, the PEP allows, or denies, the access request to the medical data.

III. Policy Authoring

This section focuses on design and implementation of an authoring tool specifically tailored for the creation and management of healthcare *privacy policy*.

A *privacy policy* consists of a set of rules expressed in terms of the four policy elements *subject*, *object*, *action*, and *environment*. Rules may represent *authorizations*, if they allow the *subject* to execute the *action* on the *object* in the *environment*. They may also represent *prohibitions*, with the obvious meaning. The policy elements are identified through their *attributes* defined according to the reference scenario.

Hereafter, we consider the healthcare scenario with the following attributes: (i) subjects can be identified through their *ID*s and *Role*s (*e.g., General Practitioners*, *Specialist*, etc.), *Organization*s which they belong to, and *Location*s, *i.e.,* phys-

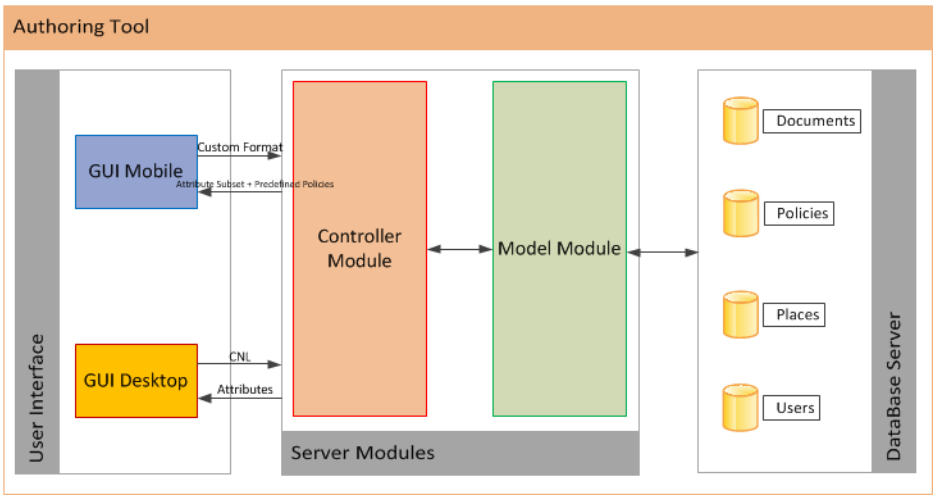
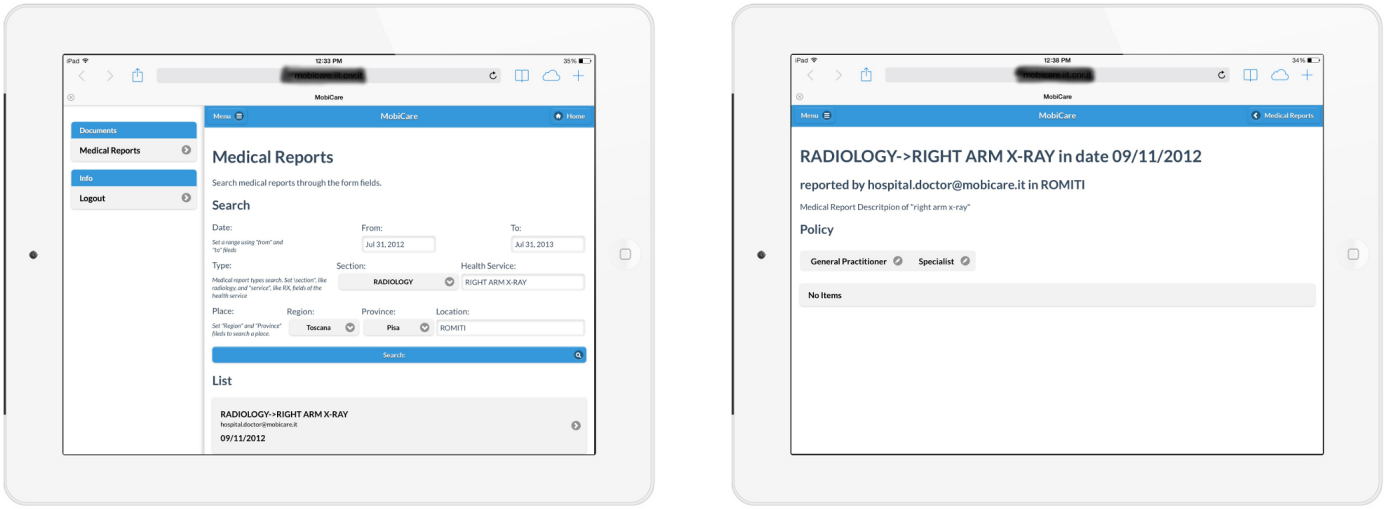


Fig. 2. The authoring tool architecture



(a) Object selection

(b) Subject selection

Fig. 3. Object and Subject selection

ical addresses where subjects are located at the time of the access request; (ii) the element object is the data over which subjects are trying to perform some action. Attributes for such data could be their *IDs*, the *Category*, *e.g., medical* or *administrative* data, their *Owners*, *e.g.,* the patient such data refer to, their *Issuers*, *i.e.,* entities that have produced those data, and the data *Location*; (iii) the *environment* can be specified through the attributes *Time* and *Date*.

The architecture of the authoring tool is shown in Figure 2. It has been designed according to the three-tier paradigm, where the three levels are: 1) the User Interface (Mobile + Desktop), developed in HTML and Javascript; 2) the internal engine, made of the Server Modules, the Controller, and the Model Module, implemented using the PHP language; and 3) a Relational Database, developed as a MySQL database server. The user interface, the controller, and the model module have been designed according to the *Model-view-controller* (MVC) pattern [4] which separates the representation of information from the user's interaction with it. The view consists of the user interface. The controller mediates inputs converting them to commands for the model or view. The model interacts with the controller and the database, by querying the latter according to which form the user fills at the interface.

The database has the following structure: *Users* schema, that contains user tables linking policy subjects to their attributes, *e.g.,* their roles, and tables linking subject attributes to their values, *e.g.,* General Practitioner; *Documents* schema, that contains tables linking policy objects to their attributes, *e.g.,* their categories, and tables linking object attributes to their values, *e.g.,* medical; *Places* schema, consisting of tables of environmental attributes, like time and date, and tables linking attributes to their values; *Policies* schema, storing the authored policies in a XACML-fashion language. This schema also stores policy actions.

The User Interface actually consists of two interfaces, the desktop one, thought for policy experts, and the mobile one, for common users with no technical skills on policy specification. The Desktop interface, for each policy element, retrieves the whole set of attributes available in the Data Base interacting with the Server Modules. The graphical interface helps the

user to combine these attributes with the proper operation in order to produce a policy rule. The Mobile interface, instead, retrieves a set of predefined policy skeletons, along with a restricted set of attributes that can be exploited to instantiate those skeletons. The policy resulting from the choices performed by the user though the interface is stored in the Data Base. The Desktop interface produces policies expressed in controlled natural language, CNL4DSA [5], while the preferences authored through the Mobile interface are stored in the Data Base using a custom format. Then the policies are mapped to XACML policies (step (4) of Figure 1) and stored in the Policy Repository.

A. User Interface

The User Interface has been designed and implemented considering that 1) users are classified according to their expertise; 2) the interface provides different sets of features according to the user expertise; and, 3) the interface is accessible by mobile phones, tablets, and desktops. As in [6], two users categories are considered:

Common Users, *e.g.,* patients or doctors that produced the medical documents, that are unaware of the constraints they can impose on their data. These authors are driven in the authoring phase through the *Mobile Interface*, that 1) is *document-centric*, *i.e.,* it allows users to compose their privacy preferences over specific documents they own in few clicks; 2) it offers a simple and guided way to compose such preferences; and, 3) is accessible from smart-phones and tablets.

Policy Experts, with a high-level understanding of the policy domain. These authors may be driven in the authoring phase through the *Desktop Interface*. The policy experts are not expected to have in-depth technical knowledge of how the policy will be evaluated and enforced, but they are familiar with high-level policy specification languages. Examples of policy experts are policy makers of national healthcare systems, that assess standard guidelines for access control and usage of sensitive medical data in their countries. The desktop interface has been especially thought for them, since, reasonably, setting the high level privacy policies fixed by national healthcare systems and healthcare organisations is an activity carried out during ordinary workdays.

1) Mobile Interface: Some screenshots of the mobile interface are shown in Figures 3 and 4. Designed for non expert people, possibly ignoring technical aspects of policy specification, its design is minimalist, to reduce the cognitive load of the user. Commands are grouped in a sliding panel on the left side of the screen, see Figure 3 a). The menu is retractable to leave space to the content that, in this way, appears not to be crushed and it is usable at different resolutions. The bar at the top of the screen allows the user to return to the homepage and to previously visited pages, and to open the panel menu (on which there is the logout button).
The Mobile Interface is *document-centric* because it allows the user to set privacy preferences on documents by firstly selecting such documents. Filling the form in Figure 3 a), the user obtains a list of the documents for which she is allowed to edit access preferences. In particular, patients are allowed to set the access preferences of their medical document, while doctors are allowed to set the access preferences of the

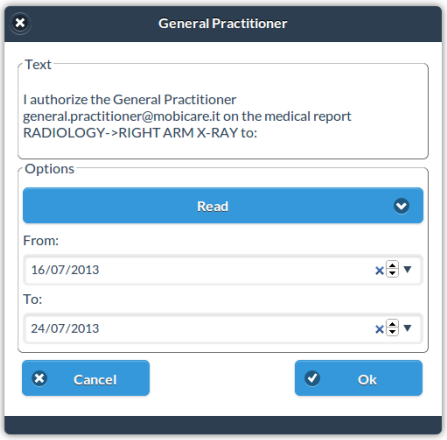


Fig. 4. Action and Environment selection

document they produced. The visualization of such list can be constrained by requiring to visualize only those documents issued within a certain time interval, or on a certain date, or of a certain category (*e.g.,* only radiological reports). Constraints on how to visualize the document list are enacted by selecting specific values from an autocomplete input.
Pairing users with the list of documents available to be visualized is possible through a two-step phase of authentication and authorization. First, a user logs into the interface by presenting her own credentials. Once logged, the system automatically retrieves the set of profiles associated to the user. Each profile represents a set of attributes paired with the user. For example a given user could have two profiles: the profile *patient*, which includes the role attribute, whose value is, obviously, *patient*, and the profile *doctor*, which includes the role attribute, with value *Psychiatrist*, and the Organization attribute with value *Psychiatric Hospital ABCD*. The profiles are defined by the entities that issue the users attributes. A user with more than one profile, *e.g., patient* and *doctor*, must choose to use the interface selecting one of them. Selecting the profile *patient*, the user will be able to edit preferences only on medical documents regarding herself as a patient. Instead, the same user, which selects the profile *doctor*, will be able to compose preferences over all the medical documents she issued, although these documents refer to different patients.
Upon document selection, the interface shows to the user different buttons associated to commands that encode partly customizable authorizations. Consider a user that has the role *patient*. First, she selects the document over which she may want to compose her privacy preferences. This document becomes the object of the policy. As an example, in Figure 3 a), she searches and selects the medical report "Right Arm X-Ray 09/11/2012". Then, she chooses the subjects of her policy, *i.e.,* the subject to whom this rule will be applicable (see Figure 3 b), in which she can select either subjects with role *General Practitioner* or *Specialist*). Upon subject selection, she can further select 1) the kind of action that the subject is allowed to perform on the object, *e.g.,* "read"; and 2) the temporal validity of the authorisation (Figure 4). In such a way, the user sets the following policy: *My General Practitioner can read my medical report "Right Arm X-Ray" from 16/07/2013 to 24/07/2013*, which is saved when he presses "Ok".

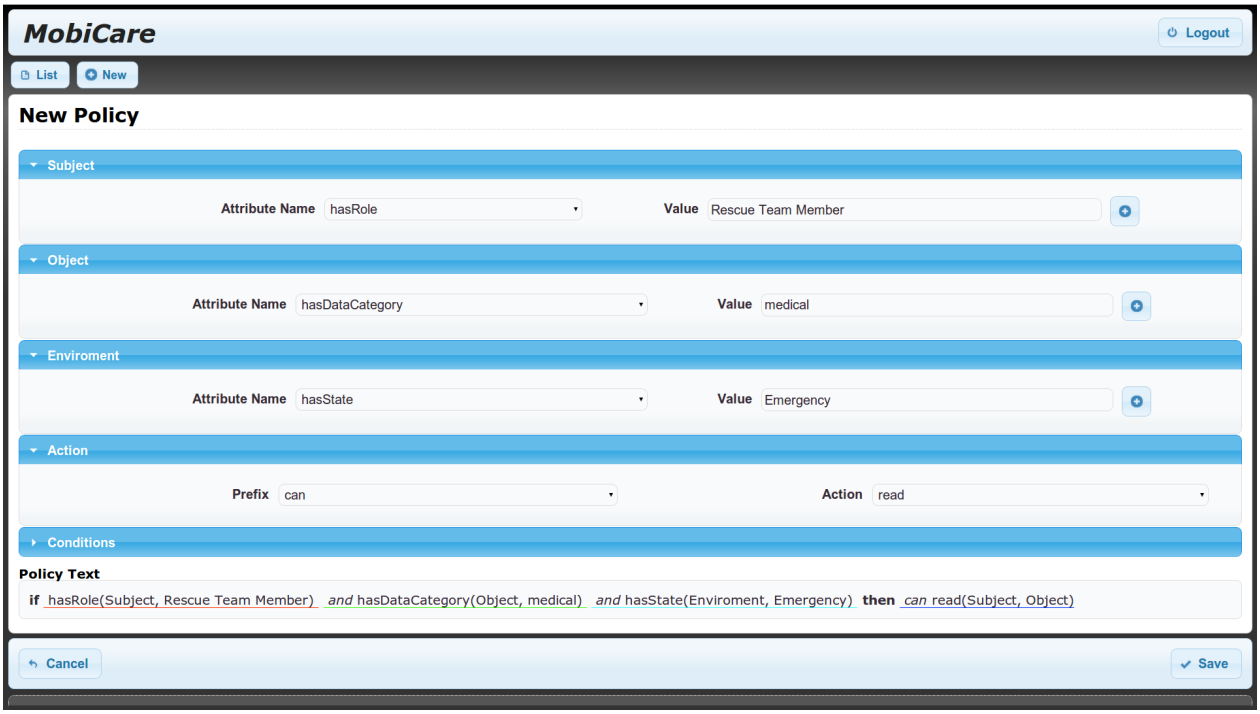


Fig. 5. Desktop Interface

2) Desktop Interface: The Desktop Interface is shown in Figure 5. It allows the editing of privacy policies in a more complex way with respect to simple setting of privacy preferences provided by the mobile interface. Its usage is reserved to Policy Experts (see above in this section). This interface presents four tabs, one for each policy element (*subject, action, object*, and *environment*), plus one tab labelled *conditions*. The latter drives the user to set comparisons between attributes. For each element, the users can select from a drop-down menu which attributes to set and the attributes values (from another menu). As an example, let the reader suppose that the user aims at composing the following authorization: *"The rescue team member can read any medical report in emergency situations"*. First, she selects the subject attribute *role* from the drop-down menu. Second, she selects the values for this attributes, *i.e., rescue team member*. Then, she selects the object attributes *category*, plus their values, *medical*. The same procedure is applied to specify action and environment attributes. To add, or remove, an attribute for the same element, there are the plus and minus buttons, respectively, at the end of value field.
The *conditions* tab allows to refine the policy by adding comparisons between attributes of the elements that constitute the policy. The drop down menus propose only the attributes that have been set in the previous tabs. The resulting policy is shown in a text box located under the tabs, and it is expressed exploiting the controlled natural language CNL4DSA [5].
In a real environment, policy makers are supposed to set a not negligible number of policies regulating the management and sharing of all the data produced and stored at their healthcare organizations. Thus, it becomes probable to have two, or more, policies that would apply to the same access request and return different results (*e.g.,* one policy denies the access to the requester, while the other policy allows it). In order to avoid the co-existence of conflicting policies among all the policies set by the policy makers through the desktop interface, we support the editing phase with a methodology for policy validation [7], aiming at detecting conflicts among a policy set of CNL4DSA privacy policies (step (1b) in Figure 1). The analysis process allows to detect conflicts between policies by performing pairwise analysis over all pairs of authorisation and prohibition clauses. The validator exhaustively simulates all the possible access requests, under a set of contextual conditions defined by the policy expert (*e.g.,* she can set date and time of the access request, role of subject, category of data, etc.). Thus, the validator checks if there exist, at least, one authorisation and one prohibition that, simultaneously, allows and denies the same subject to perform the same action on the same object, under the given set of contextual conditions. The analysis result is showed to the policy maker through a graphical user interface in such a way that she is able to eventually modify some policies to avoid conflicts among them.

IV. Related Work

Series of work in [8], [9], [10], [11], [6] connect policy authoring tools with the capability of common users to use them. In [8], the authors carry out a laboratory evaluation of a variety of user-centric methods for privacy policies authoring, to identify which design decisions should be taken for flexible and usable privacy enabling techniques. Work in [9] continues this line of research, by providing a parser which identifies the privacy policy elements in rules entered in natural languages: identification of such elements is a key step for subsequent translation of natural sentences in enforceable constructs (such as the XACML language [3]). Authors of [10] recall security and privacy policy-authoring tasks in general, and discover further usability challenges that policy authoring presents. In [11], the authors present the Coalition Policy Management

Portal for policies authoring, verification, and deployment, with the goal of providing "easy to use mechanisms for refining high-level user-specified goals into low-level controls". These works show an implementation of a prototype architecture called SPARCLE (Server Privacy Architecture and Capability Enablement). This aims at helping privacy professionals to create policies in natural language and translate those into system readable commands. The interface is designed for a desktop usage that provides a syntax guide for writing policies in natural language. However, basic users with no technical expertise could face some difficulties with this framework. For that reason, the authoring tool here proposed tries to reduce the technical skills needed to compose a policy, and offers the mobile interface to edit a privacy preference in few clicks.

Recently, work in [6] advances the notion of templates-based authoring tools, for users with different roles and skill sets, as, *e.g.,* patients, doctors, and IT administrators could be in the e-health scenario. Thus, the authors propose different templates to edit privacy policies, each of them needing different user skills, and they model a prototype interface, still not implemented. The mobile interface proposed here allows to compose preferences that will be automatically, and transparently, mapped into XACML, [3], the well known language for access control policies, sufficiently general to be put also to the purpose of privacy rules specification. Users can set the contextual conditions, like time range or locality, through drop-down or autocomplete inputs of a form. On the other hand, the desktop interface helps users to write policies using a controlled natural language (*cfr.* CNL4DSA [5]) that will be transparently mapped to XACML too.

The FP7-EU project *Consequence* designed and developed an integrated framework for the authoring, analysis, and enforcement of Data Sharing Agreements (DSA), that are formal documents regulating data exchange. The authoring tool developed within the project was intended for users with deep knowledge on policy specification [12], [13]. The use of a controlled natural language (CNL4DSA) and the insertion of a help-on-line facility partly mitigate usability issues, whose complete solution needs however further investigation. Furthermore, the Consequence authoring tool does not allow multi-party definition of policies. On the contrary, the architecture of the authoring tool presented in this paper allows different users to write policies. Whenever an access request is sent, all the policies that can be applied are taken into account and evaluated at run-time in such a way to solve conflicts among them (by exploiting, *e.g.,* the conflict detector engine provided by the XACML framework, as well as strategies for conflict resolution, such as the ones proposed in [14], [15], [16]).

From a business perspective, Axiomatics offers a desktop interface for policy authoring [17]. The GUI provides support to IT administrators with relevant technical skills to edit XACML policies. From a social networking perspective, work in [18] presents a *collaborative* authoring tool, allowing several individuals to specify policies over data published on social networks, and whose disclosure may affect their privacy. The authors acknowledge some usability issues in their prototype implementation, and future work are foreseen towards a user-friendly authoring interface. The proposed tool approaches the problem from a different prospective. Indeed, the interface allows users to write policies on own data by the setting of an authorization to other users and without involving their data (data centric model). Furthermore, different entities that specify policies on a same data are unaware of the existence of policies written by others. All the policies are analyzed at run-time after each access request to the data.

V. Conclusions and Future work

In this paper, we presented the prototype of an expertise-driven authoring tool for privacy policies, specifically tailored for the healthcare scenario. The mobile interface of the tool allows common users, such as patients, to edit their privacy preferences for a controlled management of their medical data. The desktop interface, instead, is specifically developed for policy makers that set up general policies, as dictated, *e.g.,* by national healthcare institutions. As ongoing work, a series of tests to verify the tool usability are being set up. We are currently selecting people with different expertise to carry out the tests. User feedback for refining the authoring tool and improving the user experience are the expected outcome of the testing phase. We plan to integrate the authoring tool into a policy-based infrastructure described throughout the paper.

References

[1] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The Ponder policy specification language," in *POLICY*. Springer, 2001, pp. 18–38.

[2] J. Jajodia, P. Samarati, and V. Subrahmanian, "A logical language for expressing authorizations," in *IEEE Symposyum on Security and Privacy*, 1997, pp. 31–42.

[3] OASIS, "eXtensible Access Control Markup Language (XACML) Version 3.0," January 2013.

[4] G. E. Kransner and S. Pope, "Cookbook for using the Model-View-Controller User Interface paradigm," *Object Oriented Programming*, pp. 26–49, 1988.

[5] I. Matteucci, M. Petrocchi, and M. L. Sbodio, "CNL4DSA: a Controlled Natural Language for Data Sharing Agreements," in *SAC: Privacy on the Web Track*. ACM, 2010.

[6] M. Johnson *et al.*, "Optimizing a policy authoring framework for security and privacy policies," in *SOUPS*. ACM, 2010, pp. 8:1–8:9.

[7] F. Martinelli *et al.*, "A Formal Support for Collaborative Data Sharing," in *CD-ARES*, 2012, pp. 547–561.

[8] J. Karat *et al.*, "Designing Natural Language and Structured Entry Methods for Privacy Policy Authoring," in *INTERACT*, 2005, pp. 671–684.

[9] C. Brodie *et al.*, "An Empirical Study of Natural Language Parsing of Privacy Policy Rules using the SPARCLE Policy Workbench," in *SOUPS*. ACM, 2006.

[10] R. W. Reeder *et al.*, "Usability challenges in security and privacy policy-authoring interfaces," in *INTERACT*. Springer, 2007, pp. 141–155.

[11] C. Brodie *et al.*, "The Coalition Policy Management Portal for Policy Authoring, Verification, and Deployment," in *POLICY*, 2008, pp. 247–249.

[12] Consequence Project, "Infrastructure for data sharing agreements," in *http://goo.gl/is7cpR*, Dec. 2010.

[13] I. Matteucci, M. Petrocchi, M. L. Sbodio, and L. Wiegand, "A design phase for data sharing agreements," in *DPM/SETOP*, 2011, pp. 25–41.

[14] I. Matteucci, P. Mori, and M. Petrocchi, "Prioritized Execution of Privacy Policies," in *DPM/SETOP*, 2012, pp. 133–145.

[15] A. Lunardelli, I. Matteucci, P. Mori, and M. Petrocchi, "A Prototype for Solving Conflicts in XACML-based e-Health Policies," in *Computer-Based Medical Systems*. IEEE, 2013.

[16] J. Jin *et al.*, "Patient-centric authorization framework for electronic healthcare services," *Computers & Security*, vol. 30, no. 2-3, pp. 116–127, 2011.

[17] Axiomatics.com, "Policy Administrator Point," in *http://goo.gl/A5OEHW*, last checked Jan 17, 2014.

[18] R. Wishart *et al.*, "Collaborative Privacy Policy Authoring in a Social Networking Context," in *POLICY*. IEEE, 2010, pp. 1–8.