A Semantic Policy Management Environment For End-Users

Joachim Zeiss, Rene Gabner, Anna V. Zhdanova, Sandford Bessler
(ftw. Telecommunications Research Center Vienna, Austria
{zeiss, gabner, zhdanova, bessler}@ftw.at)

Abstract: We describe a policy management environment (PME) for the Semantic Web and show its added value compared to existing policy-related developments. In particular, we detail a part of the PME, the policy acquisition tool that enables non-expert users to create and modify semantic policy rules. The implementation of such a policy editor has at its core a semantic reasoner operating on N3 rules and a simple web-based user interface. We describe applications in which PME is used and discuss the feasibility and advantages of ontology-based and community-driven policy management.

Keywords: user-generated content, policy web, semantic policies, privacy
Categories: M.5 - Knowledge Personalization and Customization, H.3.5 - Online Information Services

1 Introduction

Nowadays, community-driven Web services and portals such as Facebook, 43things.com, SecondLife, YouTube, LinkedIn and Orkut, a.k.a. Web 2.0 developments are at their popularity peak attracting millions of members. However, the existing Semantic portals and community sites [Davies et al. 2002, Karat et al. 2006, Zhdanova 2008], while collecting large amounts of user-generated content, are highly limited in providing adequate management and sharing control of the submitted data. In particular, the users normally cannot specify to whom the content or service is dedicated or under which circumstances and how it can be used. The inability to dynamically define and apply policies[1] often leads to undesired personal information disclosure, increasing amounts of electronic spam, revenue losses on licensed content and services, and inflexible information management. Due to their complexity and rather narrow scope, typical existing standalone policy solutions or platforms [XACML 2008, P3P 2008] cannot be directly employed by the end users.

A more precise knowledge representation and thus a larger degree of flexibility and adaptability in the actual policy employment can be achieved by deploying Semantic Web technologies [Berners-Lee et al. 2001, Davies et al. 2002] and community-driven ontology management [Zhdanova 2008, Zhdanova and Shvaiko 2006]. At present, Semantic Web and social software technologies are already applied in numerous community environments [Karat et al. 2006]. However, the policies and

[1] A policy is "a plan or course of action, as of a government, political party, or business, intended to influence and determine decisions, actions, and other matters" (The Free Online Dictionary).

rules on the light-weight (and often tag-based) social Web have not gained a broad usage yet, largely due to the policy acquisition bottleneck from the non-expert users.

In this paper, we argue that current Semantic social Web can gain substantial benefits from integrating community-driven policy management, i.e., enabling the community members to develop, maintain and share semantic policies.

A community-driven policy management infrastructure has to support developers and users in their efforts to create the Semantic Web policy content, i.e., designing ad-hoc policies on operating on existing ontologies. In practice, adding policy management support to applications will naturally allow more users to create, share and reuse policies on the Web, or contribute to the appearance of the "Web of Trust", extending the Semantic Web by user-generated rules.

A number of policy construction environments [Karat et al. 2006] and policy frameworks based on mark-up policy description languages such as XACML and P3P [XACML 2008, P3P 2008] have been proposed. However, none of these systems meets all the expectations of policy management for the social Semantic Web: most of these works address only narrowly restricted specific policy management functionality and underestimate the importance of community-driven policy management and shared semantics trends.

The main contributions of the presented work are:

- Definition of a user-driven PME for open, sharable infrastructures such as for Web or mobile services,
- Semantic-based implementation of the PME,
- Identification of showcases for such environment.

The paper is organized as follows. In Section 2, we describe our approach of a policy management for the social Semantic Web. In Section 3, architecture and implementation aspects are presented. Our experience with practicing community-driven policy management use cases is described in Section 4. Section 5 concludes the paper.

2 Semantic Policy Management

The following paragraphs describe the basic components of our architecture. The architecture is strongly related to conventional ontology and policy management services [Bonatti et al. 2006, Davies et al. 2002], but is enriched with end-user generated policy acquisition and advanced policy communication. The basic model is that of an open system in which policy rules can be shared, adapted to individual needs and enriched with facts and instance combinations.

A **Policy Storage and Query** component is provided to efficiently store and query parts of policy data and metadata by providing indexing, searching and query facilities for ontologies. In addition to conventional policy management services [Bonatti et al. 2006, Davies et al. 2002], we propose to enrich the existing search and query components with community-generated policy information. This would improve their performance and make the search, reasoning and consistency checking features mature and more attractive to use.

As the users of the environment are generally not bound to a single community or application, they must be able to publish personal and community-related policies in a

multi-accessible way. The current focus in semantic policy storage and querying is thus maintaining distributed repositories with functionalities for aggregation, decomposition and discovery of information in simple ways.

A **Policy Editing** component is introduced for creating and maintaining policies and instance data. The front-end, a user-friendly interface, helps users to easily add and modify policy-like rules on the basis of existing imported ontology classes and properties shared among several users and communities, policies and instances. The back-end consists of a storage and query system. A Policy Editor enables sharable editing for multiple users and tight integration with semantic publishing, delivery and visualization components, allowing the involved parties to observe the evolution of policy settings. These requirements are due to the elevated degree of flexibility required by community-oriented environments as the social Semantic Web and its members to freely evolve schemata, policies and to influence community processes.

A **Policy Versioning** component is introduced to maintain different versions of policy definitions, as communities, content and relationships change over time. The user should be able to easily adapt policies to new scenarios and communities without losing previous definitions. Earlier versions can be reused for definitions of new policies. Also users could experiment with more restricting policy definitions and roll back to previous versions wherever practical. A Policy Versioning component interacts with existing versioning systems like svn [Collins-Sussman et al. 2004] to provide a versioning service to the user. Semantic metadata describes the necessary versioning information inside the policy definition itself.

A **Policy User Profile and Personalization** component is responsible for the users' access to the environment and it connects the policies with the user profiles. At a more advanced level, the component helps to share and communicate policies across the user's profiles, apply policies depending on the user profiles and recommend policies basing on the user profiles. In particular, access and trust policies can be implemented taking into consideration community and social networking information provided by the users [Golbeck et al. 2003].

Our ***overall*** ontology-based **policy management** approach features: ***user-driven policy construction***, meaning that the system extensively assists the users to model the policies correctly (e.g., proactive suggestion of the ontology items that can be combined in a policy, consistency checking for the modelled policy solutions); ***policy semantic representation and sharing across communities,*** essential for the further extension for the rules layer of the Semantic Web; ontology import and ***policy creation on the basis of shared ontologies,*** the user is free to input any ontologies he/she likes and define policies on them.

Thus, ontology-based and community-oriented policy management is an advance over a conventional policy management. The advantages are gained by introducing an infrastructure that enables the communities to manage their policies.

3 Implementation

The implemented infrastructure is designed as a component for a community Semantic Web portal providing policy management facilities to the community members and managers. The infrastructure is built as a Web-based application using JSON technology [Crockford 2006, JSON 2007] and exploiting a Python version of

Euler [De Roo 2008] for manipulating ontology schemata, instance data and policies in an N3 format [Berners-Lee et al. 2007]. In Figure 1, the architecture consists of two major blocks, the policy engine and the policy acquisition tool (PAT). Whereas the PAT server interacts with the end-user via a web front end, the policy engine is responsible for the "logical" side of the system, accomplishing integration of external and internal information, reasoning and rule production. The PAT server sends requests to the policy engine whenever the user loads a policy, selects the policy building blocks or saves a policy. The incoming request and the user context are the only input data.

In order to develop policies for a certain application (domain), we need the availability of domain-dependent and domain-independent ontologies. We need as well service support for the portals' data and metadata, mostly, through publishing services for making human-readable the semantically-enriched data. Non-semantic data from a profile or context management system (e.g., XML structured data) is converted to a N3 format. OWL and RDF data can also be used.

The core component of the architecture is the policy engine (PE), a stateless request/response based server that deals with any kind of requests expressed in N3 [1]. The policy engine has associated a *Decision Space*, a set of files containing N3 triplets as well as rule objects, i.e., parsed N3 statements, kept in memory. The files contain persistent semantic data like ontology definitions, instance data and rules. Volatile semantic data relevant for the current policy request are added to the N3 objects in memory. The *Request Processor* is the part of the PE that extracts data from the request (out of a SIP message, a http GET/POST message or a SMS) and inserts it into the decision space. The policy engine may also extract data from a user profile, user context such as location, or policy data via an additional context interface. The *Reasoner*, the heart of PE, is a N3 rule engine that is invoked with the receipt of a request and uses all semantic data made available in the decision space as reasoning input. The reasoner is based on the python implementation of Euler (backward-chaining, enhanced with Euler path detection).

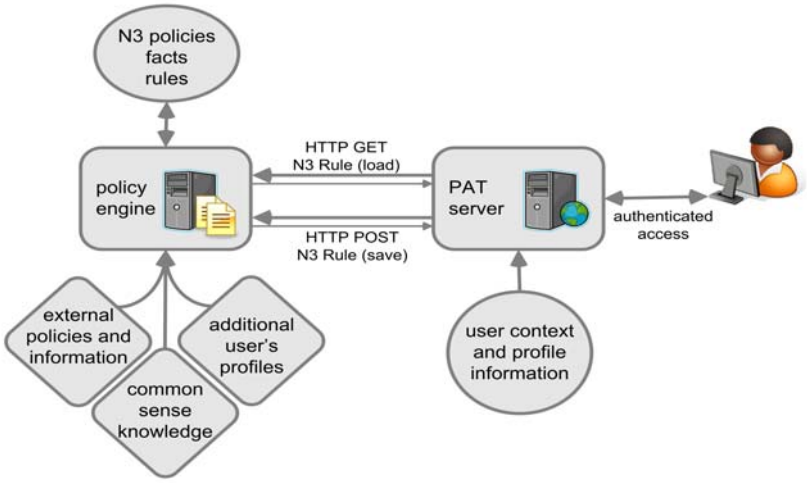

Figure 1: Policy management infrastructure

In the design of the policy acquisition tool (PAT) we have introduced certain ***novel techniques***.

The user *interface is dynamically generated* from the ontologies and the instance data. The latter are provided by the end user or deduced by the policy engine based on defined business logic. In addition to ontologies, user profiles and context data are used to assist the user in editing the policies. The implemented user interface is shown in Figure 2.

At the moment PAT offers to combine data according to profile and context ontologies. All semantic information (describing policies and user rules) is encoded in N3 format. As N3 format is triplet-based, the environment's knowledge representation is lightweight and *caters to a straightforward reuse of semantic content in other formats* (e.g., in RDF(S) and OWL).

The tool consists of a web front end that presents a JavaScript enabled user interface and a (PAT) server part that contains the logic of creating and processing a rule. The PAT server queries the policy engine every time it receives an update request from the client (and converts the JSON data received from the client into a valid N3 request and vice versa). This feature enables PAT to *provide the user only with data (subjects, predicates and objects) which are valid* from the ontology perspective to construct a consistent rule. All the rules are acquired from and stored into the policy engine's decision space.

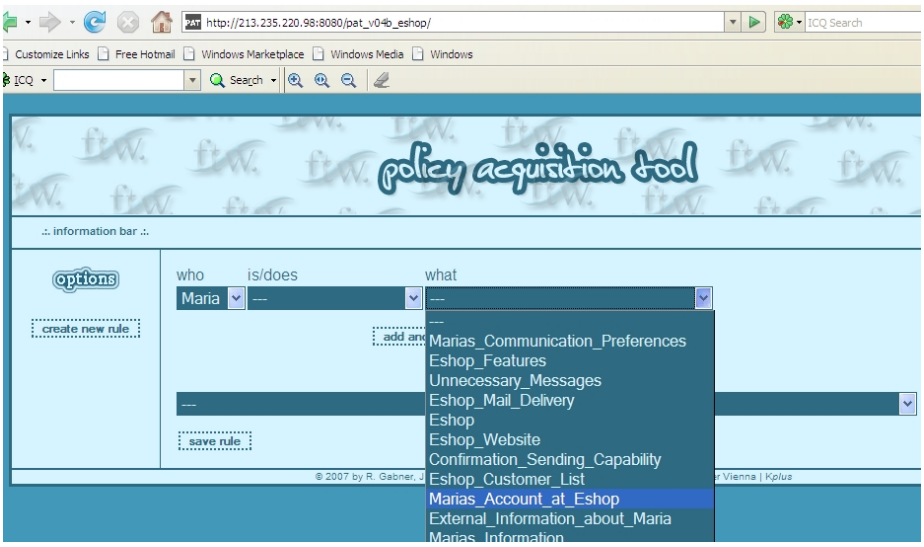

Figure 2: Graphical interface for policy acquisition

4 Applications

In this section, we describe applications of the ontology-based PME and our experiences with them.

Eshop / Internet Shopping: In this case study, the user of the PME is a manager of an internet shop (Eshop). He/she needs to model the online shop's privacy policies

in a policy editor for a group of customers. In the example below, we present such a policy for a typical customer named Maria. Maria regularly shops online, likes special offers and recommendations, but wants to keep sharing of her personal profile information under control. One of the policies; that might be valid for an Eshop and should be known to Maria and its N3 representation are as follows: "*We might receive information about you from other sources and add it to our account information*".

```
Maria a :Customer.
Eshop a :Eshop.
External_Information_about_Maria a :External_Customer_Information.
Marias_Account_at_Eshop a :Eshop_Customer_Account.
{Maria :has Marias_Account_at_Eshop.
Eshop :receives  External_Information_about_Maria
}=> {External_Information_about_Maria :is_added_to
Marias_Account_at_Eshop}
```

Mobile Service Marketplace: The policy management environment is applicable for selling services at the Web or mobile markets. The end user defines the service and sets the service descriptions and as well as policies as the conditions on provisioning or selling the service. The core service descriptions and their model are predefined in an ontology for a micro-service. Only specific areas of service definition can be overwritten or provisioned by the user employing an ontology-based policy management environment.

Policy-Driven Personal Networks: In [Zeiss et al. 2007] two or more owners of personal networks decide to interact and form a so-called federation to access the each other data or services. In this scenario our PME is being used to control setup, usage and takedown of personal network federations. It enables the user to maintain privacy and to control the access to resources and services inside his/her network, while opening up towards other private networks to use foreign services securely. A complex task made easy by introducing semantic policies and a user-friendly policy editor.

Context Aware Services: In [Bessler and Zeiss 2006] the relationship between user and provider of a context aware service is discussed. The provider needs to protect his/her resources without degrading service functionality. The user in turn is interested in protecting his/her privacy but still wants to offer sufficient context data to obtain personalized service results. Our PME helps both parties to keep the balance of contradicting interests by automating the necessary negotiation between user and provider.

User Availability: In [Bessler and Zeiss 2007] an alternative to existing presence and availability architectures is being introduced. In this use case the idea is not to control the delivery of intimate presence or location information. In fact no such information is revealed at all. A decision on if, how and when a user wants to be contacted is being communicated instead of delivering sensitive private data. In this scenario, the PME enables a communication device to automatically decide how, when and by whom a connection can be established based on context data, user profiles, buddy lists and user defined policies.

5 Discussion and Conclusions

Summarizing, we see the following value being added by an ontology-based policy management compared to conventional policy practices:

1. ***Spreading of policies***, freedom in policy distribution and sharing, annotation of the end users' data and services, easiness in reading other people and organizations' policies; all these are would be difficult without the semantic practices.
2. ***Reduction of costs for policy construction***: existing similar policies may be available and easy to reuse elsewhere. For example, most of the internet shops have very similar polices on how to deal with the customer data and they would not need to redefine all the policies from scratch. One could also advance eGovernment visions by provisioning machine readable laws, e.g., on data protection,
3. ***Reduction of the mistakes in the user-generated policy modeling*** as the system's storage, query and reasoning service as well as sharing of policies within communities act as controllers for policy correctness.
4. ***Better awareness of the end users about policies, rules and regulation:*** With the suggested system the policies are easily retrieved and presented to the users.

The evaluation of the policy management environment is being done via the user studies, i.e., by placing the system online and letting the volunteers to build policies and/or to give a feedback via a pre-designed questionnaire. Then the users' inputs and feedback is analyzed. The sample scenarios specified in Section 5 are being offered as the evaluation scenes.

Apart from technical and usability issues, the following more socially-oriented questions should be investigated in community-driven policy modeling studies:

- How users share personal data, multiple identities, etc. Initial observations can be drawn from social networking websites (e.g., LinkedIn, Xing, etc.) where users can select whether they share specific type of information with other users;
- Specifying, accumulating and storing arbitrary policies could result in a "policy Wikipedia" provisioning commonsense knowledge rules of what users find right and appropriate, e.g., "do not drink and drive". Such community effort would also have an anthropological effect in enabling observation of which kinds of policies are shared between large communities and which policies are less popular.
- Certain policies vary by countries, cultures and time (e.g., eating any kind of foods using hands could have been acceptable in certain countries in the past, but not in the present). This adds to additional technical challenges in policy versioning, matching and comparison.

We have introduced ontology-based policy management and its benefits. In addition, we have described an implementation supporting ontology-based policy management and discussed its actual and potential applications. As a conclusion, we are convinced that the ontology-based policy management is a highly important concept for services offered in user-centered open environments, such as Web or mobile environments. Also we foresee that implementations of such ontology-based policy management infrastructure will become an essential part of end-user service-oriented environments involving policies.

Acknowledgements

The Telecommunications Research Center Vienna (ftw.) is supported by the Austrian government and the City of Vienna within the competence center program COMET. This work is partially funded by the IST project Magnet Beyond (http://www.ist-magnet.org).

References

[Berners-Lee et al. 2007] Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J., 2007. "N3Logic: A Logic for the Web", *Journal of Theory and Practice of Logic Programming (TPLP)*, Special Issue on Logic Programming and the Web, 2007.

[Berners-Lee et al. 2001] Berners-Lee, T., Hendler, J., Lassila, O., 2001. "The Semantic Web". *Scientific American 284(5)*, pp. 34-43.

[Bessler and Zeiss 2006] Bessler, S., Zeiss, J., 2006. „Semantic modelling of policies for context-aware services", *Wireless World Research Form (WWRF17)*, November 2006.

[Bessler and Zeiss 2007] Bessler, S., Zeiss, J., 2007. „Using Semantic Policies to Reason over User Availability", *Proc. of 2nd Int. Workshop on Personalized Networks (Pernet07)*, 10 August 2007, IEEE Press.

[Bonatti et al. 2006] Bonatti, P.A., Duma, C., Fuchs, N., Nejdl, W., Olmedilla, D., Peer, J., Shahmehri, N., 2006. "Semantic web policies - a discussion of requirements and research issues", In *3rd European Semantic Web Conference (ESWC)*, 11-14 June 2006, Budva, Montenegro, Springer-Verlag, LNCS 4011, pp. 712-724.

[Collins-Sussman et al. 2004] Collins-Sussman, B., Fitzpatrick, B.W., Pilato, C.M., 2004. "Version Control with Subversion", o'Reilly 2004.

[Crockford 2006] Crockford, D., 2006. *The application/json Media Type for JavaScript Object Notation (JSON)*, RFC 2647, July 2006.

[Davies et al. 2002] Davies, J., Fensel, D., van Harmelen, F. (eds.), 2002. *Towards the Semantic Web: Ontology-Driven Knowledge Management*, John Wiley & Sons.

[Golbeck et al. 2003] Golbeck, J., Parsia, B., Hendler, J, 2003. "Trust Networks on the Semantic Web", In *Proceedings of Cooperative Intelligent Agents 2003*, Helsinki, Finland.

[JSON 2007] JSON, 2007. URL: http://www.json.org.

[Karat et al. 2006] Karat, C.-M., Karat, J., Brodie, C., Feng, J., 2006. Evaluating Interfaces for Privacy Policy Rule Authoring, *In Proc. of the Conference on Human Factors in Computing Systems (CHI 2006)*, pp. 83-92.

[Mika 2007] Mika, P., 2007. *Social Networks and the Semantic Web***.** Springer Verlag, 234 p.

[XACML 2008] OASIS eXtensible Access Control Markup Language (XACML) 2.0, 2008. URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[P3P 2008] Platform for Privacy Preferences (P3P) 1.0, 2008. URL: http://www.w3.org/P3P.

[De Roo 2008] De Roo, J., 2008. Euler proof mechanism, 2008. URL: http://www.agfa.com/w3c/euler.

[Zeiss et al. 2007] Zeiss, J., Sanchez, L., Bessler, S., 2007. „Policy driven formation of federations between personal networks", *Proc. of Mobile & Wireless Communications Summit*, July 2007.

[Zhdanova 2008] Zhdanova, A.V., 2008. "Community-driven Ontology Construction in Social Networking Portals", *International Journal on Web Intelligence and Agent Systems, Vol. 6, No. 1*, IOS Press, pp. 93-121 (2008).

[Zhdanova and Shvaiko 2006] Zhdanova, A.V., Shvaiko, P., 2006. "Community-Driven Ontology Matching". In *Proc. of the 3rd European Semantic Web Conference (ESWC'2006)*, 11-14 June 2006, Budva, Montenegro, Springer-Verlag, LNCS 4011, pp. 34-49 (2006).